Roll Number: _____ **Name**:_____

CS 387 Midsem

3rd March 2023 Time: 830am - 925 am

Marking scheme: Each blank carries 2 marks unless otherwise stated. Total marks: 50

1. SQL: Consider the relations students(sID, studentName) that contain the list of all the students and POR(sID, studentName, position) that contain the list of the students who have had a position of responsibility (secretary etc). Fill in each of the blanks below to get 3 queries that all output students who have never held a position of responsibility. (total 6 marks)
   a. SELECT s.sID, s.studentName FROM students s WHERE_ NOT EXISTS_
      (SELECT r.sid FROM POR r WHERE r.sid = s.sid);
   b. SELECT s.sID, s.studentName FROM students s LEFT OUTER JOIN POR r
      ON s.sID = r.sID WHERE r.sID __ IS NULL_;.
   c. SELECT sID, studentName FROM students WHERE sID NOT IN
      (SELECT ___ sID from POR__);.

2. SQL: Consider the tables below, where an order can be for multiple products, which are stored in different tuples in order_details. (Each blank is 1 mark, total 10 marks)

   orders (order_id, customer_id, employee_id)
   customers(customer_id, customer_name, country)
   order_details(order_id, order_detail_id, product_id, unit_price, quantity)
   employees(employee_id ,employee_name, salary)
   product(product_id, product_name)

   a. Fill in the blanks to find all employees who have sold more than 5000 worth of products:
      SELECT employees.employee_id, employees.employee_name
      FROM order_details JOIN orders
      ON order_details.order_id = orders.order_id
      JOIN employees ON orders.employee_id = employees.employee_id
      GROUP BY employee_id,_, _employee_name
      HAVING SUM(_unit_price * quantity_) > 5000;

   b. Fill in the blanks to find product_id never ordered by a customer from India
      SELECT product_id, product_name
      FROM product
      WHERE product_id NOT IN_ (SELECT product_id
      FROM _orders__ NATURAL JOIN _customer,__
      NATURAL JOIN _order_detail_.
      WHERE Customer.country = 'India' OR country='India')

P1- Lakshya

c. The following query was written to find all employees whose salary is greater than that of employee named Sundar

```
SELECT * FROM employee e1
WHERE e1.salary > (SELECT salary FROM employee e2
                   WHERE employee_name = 'Sundar')
```

Suppose that there is no employee with the name 'Sundar'.. The query would result in _empty result _ (choose from options: empty result, output all employees, runtime exception).
Now suppose there are two employees with the name Sundar. The query would result in _Runtime error/exception_ (choose from options below).

i. Output all employees whose salary is greater than both the employees with name Sundar
ii. Output all employees whose salary is greater than at least one of the employees with name Sundar
iii. Runtime error/exception
iv. Empty result

3. PsycoPG (1 mark per blank, 4 marks)

a. The _description_ attribute of psycopg cursor allows us to get the column names of query results set after execution.
b. The function cursor._fetchall()_ is used to obtain all the rows in the result set. Fill in just the function name.
c. Passing a parameter to an SQL query using string concatenation (e.g. query = "select * from student where id =" + ID;) creates a security risk called _SQL injection__
d. Fill in the blanks below for the safe way to avoid the above problem:
cur.execute("select * from student where id = __%s__", ID);

4. JS/Node.js: (marks 6)
   a. Fill in the blank to ensure that value V is not a promise
      val V = __await__ A();
   b. Suppose functions A() and B() are asynchronous (i.e. return a promise) but function C() must return a value, not a promise. The two alternatives from below that do this correctly are __B__ and __D__

```
A) async function  C() {
        const valA = A();
        const valB = B();
        valC = valA + valB;
        return valC;
}
B) async function  C() {
        const valA = await A();
        const valB = await B();
        valC = valA + valB;
        return valC; }
```

Roll Number: _____ **Name**:_____

C) async function C() {
const valA = async A();
const valB = async B();
valC = valA + valB;
return valC;
}
D) async function C() {
const valA = A();
const valB = B();
valC = await (valA + valB);
return valC;
}
There might be multiple correct options.

5. Node.js and HTTP (1 mark last two blanks, total 10 marks)
   a. Fill in the blanks to check if the user is logged in (using expressJS sessions)

```
app.get('/',(req,res) => {
    session= req.session;
    if(session.user){  // logged in
      …
    }
    else { // not logged in, use parameters userid and password in req
          // to validate and update session
       userid = __req.body.email,_
       password = _req.body.password_
       If (user and password are validated) {
         _session.user_ = userid;
       }
  }
```

   b. Session attributes are implemented by storing a random string in Cookies in browser_ (choose from backend, cookies in browser, hidden attribute in form) and storing attribute values in the Backend_ (choose from backend, cookies in browser, hidden attribute in form).

6. Spark Short answers (1 mark per blank for a,b and 2 marks for c, d, total 10 marks)
   a. Spark _DataFrames_ support structured data processing by providing an interface for manipulating data in tabular form just like a relational algebra.
   b. The _ collect() __ function is used to retrieve all data from RDD in an Array of Row type. Fill in just the function name.
   c. The function_map__ transforms an RDD of length N into another RDD also of length N, while _ flatMap, _ transforms an RDD of length N into another RDD of length M where N and M may be different.
   d. Given an input RDD words, with each entry containing a pair of words, fill in the lambda functions below to get the count of each pair?
   words.map(lambda x: _(x,1)_).reduceByKey(lambda a, b:_a+b _)

P3 - Parismita

7. React: (1 mark per blank, total 4 mark)
    a. State in React refers to ___Internal storage__ (choose from Database state, Internal storage of the component, External storage of the component and None of the above.
    b. As the state of the React component is changed, fill in the blanks to indicate which all of the following will happen:
        i. Nothing will happen; you have to call the render method to render the component again.
        ii. The component will be automatically rerendered.
        iii. You have to create the component all over again.
        iv. The value is saved in the database
    c. ___Props__ is used to pass data to a component from outside in React.js (choose from SetState, Render with arguments, Props, PropTypes)

    d. __this.setState__ is used to change the state of the React.js component (choose from this.setState, this.setChangeState, this.State{}=..., None of the above).

P4 - Deeksha

Roll Number: ____________________ **Name**:____