**1. Suppose you’ve created a table called ‘menu’ using below SQL query:**

CREATE TABLE menu ( dish\_id SERIAL PRIMARY KEY, name varchar);

Now, I want to insert some records in the table menu:

INSERT INTO menu (name) VALUES('Fish');

INSERT INTO menu (name) VALUES('Beef');

INSERT INTO menu (name) VALUES('Veggie');

INSERT INTO menu (name) VALUES('Mushrooms');

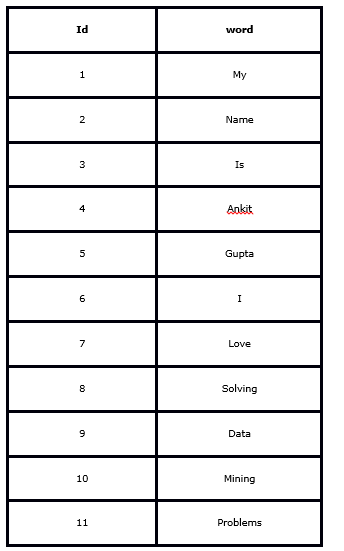
Which of the following will be the output of the below query?

Select \* FROM menu;

This will select all the available fields in “menu”:

|  |  |
| --- | --- |
| dish\_id | name |
| 1 | Fish |
| 2 | Beef |
| 3 | Veggie |
| 4 | Mushrooms |

**2. Suppose you are given a table ‘words’. The table has 2 columns ‘id’ and ‘word’.**



What will be the output for the below query?

select c1, c2, c3 from

( select id, lag(word) over (order by id) as c1,

word as c2,

lead(word) over (order by id) as c3

from words ) as t

where c2 = ‘Mining’ or c2 = ‘Problems’;

|  |  |
| --- | --- |
| **10** | **NULL** |
| **11** | **Mining** |
| **10** | **Problems** |

**3. Suppose you have a CSV file which has 3 columns (‘User\_ID’, ‘Gender’, ‘product\_ID’) and 7150884 rows. You have created a table “train” from this file in SQL.**

Now, you run Query 1 (mentioned below):

EXPLAIN SELECT \* from train WHERE product\_ID like '%7085%';

Then, you created product\_ID columns as an index in ‘train’ table using below SQL query:

CREATE INDEX product\_ID ON train(Product\_ID)

Suppose, you run Query 2 (same as Query 1) on train table.

EXPLAIN SELECT \* from train WHERE product\_ID like '%7085%';

Let T1 and T2 be time taken by Query 1 and Query 2 respectively. Which query will take less time to execute?

Both should take similar time because of SELECT \*

**4. Indexing is useful in a database for fast searching. Generally, B-tree is used for indexing in a database.**

Now, you want to use Binary Search Tree instead of B-tree.

Suppose there are numbers between 1 and 100 and you want to search the number 35 using Binary Search Tree algorithm. Which of the following sequences CANNOT be the sequence for the numbers examined?

A. 10, 75, 64, 43, 60, 57, 55

B. 90, 12, 68, 34, 62, 45, 55

C. 9, 85, 47, 68, 43, 57, 55

D. 79, 14, 72, 56, 16, 53, 55

**5. Consider the following relational schema.**

Students(rollno: integer, sname: string)

Courses (courseno: integer, cname: string)

Registration (rollno: integer, courseno: integer, percent: real)

Now, which of the following query would be able to find the unique names of all students having score more than 90% in the courseno 107?

A. SELECT DISTINCT S.sname FROM Students as S, Registration as R WHERE R.rollno=S.rollno AND R.courseno=107 AND R.percent >90

B. SELECT UNIQUE S.sname FROM Students as S, Registration as R WHERE R.rollno=S.rollno AND R.courseno=107 AND R.percent >90

C. SELECT sname FROM Students as S, Registration as R WHERE R.rollno=S.rollno AND R.courseno=107 AND R.percent >90

D. None of these