

Homework 1

Important note: All homework is to be done individually -- you are not to work with others on it. **Total Points: 10**

Question (a-j): 1 points

- a. Can SQL be used as a general-purpose programming language? a) yes b) no, it is specifically designed for database querying
- b. What is another word for “table”? instance or relation instance
- c. Is SQL a procedural language or a declarative language? a) procedural b) declarative
- d. Fill in the blank: The ‘value’ of a programming language variable is equivalent to a relation instance as the ‘type’ of a programming language variable is equivalent to a schema.
- e. Fill in the blanks: if attribute “student_id” is the primary key of a table, then no two rows of the table can have the same value of attribute.
- f. What is the name of the standard Java API to a database? Java Database Connectivity (JDBC) API.
- g. Suppose a relation schema has attributes {a,b,c,d,e}. If {a,b} is a candidate key for the schema, then is {a,b,d} also a candidate key? (yes or no). {a,b,d} contains an unnecessary attribute (i.e., d), which would make {a,b,d} not fall under the candidate key definition.
- h. Can a relation schema have two primary keys? (yes or no). There can only be one primary key.
- i. An SQL ‘create table’ statement can contain one or more foreign key definitions. This foreign key definition of table ‘takes’
foreign key (ID) references student
says that every value in the ID column of ‘student’ must match a value in table

'student'. What is the column of the matching value in 'student': a) column 'ID', b) the primary key column of 'student', or c) any column of 'student'.

j. Write pseudocode to search a node in Binary tree

```
Class Node
{
    Object key; //value of this node
    Node leftChild; //left child of this node
    Node rightChild; //right child of this node

    Object getKey() {return key;}

    Node getLeftChild() {return leftChild;}

    Node getRightChild() {return rightChild;}
}

Class BinaryTree
{
    Node root; //root of the binary tree
    Object key; //desired value to find in binary tree

    Node binarySearch(Node currentNode, Object key)
    {
        if(currentNode.getKey() == null || currentNode.getKey() == key) //if value of currentNode is null or equals to key, return currentNode
            return currentNode;
        else if(currentNode.getKey() < key) //else if key is less than current node, recurse with left child
            binarySearch(currentNode.getLeftChild());
        else //currentNode.getKey() > key //else if key is more than current node, recurse with right child
            binarySearch(currentNode.getRightChild());
        }
    }
```

k. Submit the proof (snapshot of transcript) showing SJSU ID and prerequisite CS146 with grade C- or better.