1. What makes SQL a nonprocedural language?

**SQL is a nonprocedural language because it determines what should be done. Databases must implement the SQL requests that the user creates.**

1. How can you tell whether a database is truly relational?

**Apply Dr. Codd's 12th rule. Unique keys for tables.**

1. What can you do with SQL?

**Using SQL, a user can edit, and modify information in any database. The user can set security measures and permissions on databases.**

1. Name the process that separates data into distinct, unique sets.

**Normalization reduces the amount of repetition and complexity of the structure of the previous level.**

1. Do the following statements return the same or different output:

SELECT \* FROM ARRESTS; select \* from arrests;

**Same output.**

1. None of the following queries work. Why not?

select \*;

**FROM clause missing.**

Select \* from checks

**semicolon missing**

Select amount name payee FROM checks;

**need comma between each column name**

1. Which of the following SQL statements will work?

select \* from checks; select \* from checks; select \* from checks /

**All of the above work.**

Given the following table description for the arrests table:

|  |  |  |
| --- | --- | --- |
| nysid | officerId | topCharge |

Do the following:

1. Write a query to return just the check officerId and the topCharge.

**SELECT FROM officerId# , SELECT FROM topCharge;**

1. Rewrite the query from exercise 1 so that the topCharge will appear as the first column in your query results.

**SELECT FROM topcharge, officerid# from officerId**

1. Using the arrests table, write a query to return all the unique topCharges.

**SELECT DISTINCT topCharges FROM officerId**

Use the doubleAgents table to answer the following questions.

| **LASTNAME** | **FIRSTNAME** | **AREACODE** | **PHONE** | **ST** | **ZIP** |
| --- | --- | --- | --- | --- | --- |
| BUNDY | AL | 100 | 555-1111 | IL | 22333 |
| MEZA | AL | 200 | 555-2222 | UK |  |
| MERRICK | BUD | 300 | 555-6666 | CO | 80212 |
| MAST | JD | 381 | 555-6767 | LA | 23456 |
| BULHER | FERRIS | 345 | 555-3223 | IL | 23332 |
| PERKINS | ALTON | 911 | 555-3116 | CA | 95633 |
| BOSS | SIR | 204 | 555-2345 | CT | 95633 |

1. Write a query that returns everyone in the database whose last name begins with M.

**SELECT \* FROM FRIENDS WHERE LASTNAME LIKE ‘M%’;**

1. Write a query that returns everyone who lives in Illinois with a first name of AL.

**SELECT \* FROM FRIENDS, WHERE STATE = ‘IL’ AND FIRSTNAME = ‘AL’;**

1. What shorthand could you use instead of WHERE a >= 10 AND a <=30?

**WHERE a BETWEEN 10 AND 30;**

1. What will this query return?

SELECT FIRSTNAME FROM DOUBLE\_AGENTS WHERE FIRSTNAME = 'AL' AND LASTNAME = 'BULHER';

**Nothing will be returned, none of the conditions are true**

1. Using the DOUBLEAGENTS table, write a query that returns the following:

| **NAME** | **ST** |
| --- | --- |
| AL FROM | IL |

**Select (FIRSTNAME || ‘FROM’) NAME, STATE, FROM FRIENDS, WHERE STATE = ‘IL’, AND, LASTNAME = ‘BUNDY’;**

1. Using the DOUBLEAGENTS table, write a query that returns the following:

| **NAME** | **PHONE** |
| --- | --- |
| MERRICK, BUD | 300-555-6666 |
| MAST, JD | 381-555-6767 |
| BULHER, FERRIS | 345-555-3223 |

**SELECT LASTANAME || ‘,’ || FIRSTNAME NAME, AREACODE || ‘,’ || PHONE PHONE, FROM FRIENDS, WHERE AREACODE BETWEEN 300 AND 400;**

1. Which function capitalizes the first letter of a character string and makes the rest lowercase?

**INITCAP**

1. Which functions are also known by the *same* name?

**Aggregate** **functions**

1. Will this query work?

SELECT **COUNT**(LASTNAME) FROM CHARACTERS;

**-Yes, it will return the total of rows**

1. How about this one?

SELECT SUM(LASTNAME) FROM CHARACTERS

**Nope, LASTNAME is a character field**

1. Assuming that they are separate columns, which function(s) would splice together FIRSTNAME and LASTNAME?

**| | or CONCAT function**

1. What does the answer 37 mean from the following SELECT?

SELECT COUNT(\*) FROM drone\_strikes;

**37 is the number of records in the table**

1. Will the following statement work? (Hint: look up substr)

SELECT SUBSTR LASTNAME,1,5 FROM NAME\_TBL;

**Nope, you are missing () around “LASTNAME,1,5”**

Marksmanship table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| officerId | FirstName | LastName | hits | shotsTaken |

1. Using a table called SHOOTSTATS table, write a query to determine who is are on target less than .25.

**USE SHOOTSTATS; GO SELECT officerID, Name FROM FirstName WHERE hits < .25;**

1. Using today's OFFICERS table, write a query that will return the following:

officers table

| **First** | **Middle** | **Last** | **BadgeID** |
| --- | --- | --- | --- |
| Kevin | Anthony | Petrone | 32 |

OUTPUT:

| **INITIALS** | **CODE** |
| --- | --- |
| K.A.P. | 32 |

Which clause works just like LIKE(%)? (HINT: Look it up on google.)

**Starting With**

1. What is the function of the GROUP BY clause, and what other clause does it act like?

**The function of the GROUP BY clause groups result sets from the data. The function acts like ORDER BY.**

1. Will this SELECT work?

NAME, AVG(SALARY), DEPARTMENT FROM PAY\_TBL WHERE DEPARTMENT = 'SWAT' ORDER BY NAME GROUP BY DEPARTMENT, SALARY;

**No, GROUP BY must come before ORDER BY**

1. When using the HAVING clause, do you always have to use a GROUP BY also?

**Yes**

1. Can you use ORDER BY on a column that is not one of the columns in the SELECT statement?

**Yes**

1. Using the ORGCHART table from the following examples, find out how many people on each team have 30 or more days of sick leave.

Here is your baseline that shows how many folks are on each team.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| empId | First | Last | Team | Sickleave |
| 1 | Alan | Turing | Algebra | 31 |
| 2 | John | Von Neuman | PDE | 32 |
| 3 | Robert | Oppenhiemer | Physics | 27 |
| 4 | Enrico | Fermi | Physics | 24 |
| 5 | Leo | Szilard | Physics | 37 |
| 6 | George | Danzig | Operations | 22 |
| 7 | Eric | Djkstra | CS | 21 |
| 8 | Linus | Torvals | CS | 36 |
| 9 | Richard | Stallman | CS | 40 |

Compare it to the query that solves the question: INPUT:

SELECT TEAM, COUNT(TEAM) FROM ORGCHART WHERE SICKLEAVE >=30 GROUP BY TEAM;

**-The output shows the number of people on each team with a balance of 30 days or more labeled SICKLEAVE**