Oleg Kobec

CSCI 391

Homework 2

1. What makes SQL a nonprocedural language?  
  
A procedural language requires that you define how to arrive at the answer, a non-procedural language allows you to specify what answer you want. So SQL is non-procedural.

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

Codd’s 12 Rules for a Truly Relational Database System

1. A relational DBMS must be able to manage databases entirely through its relational capabilities.
2. Information is represented logically in tables.
3. Data must be logically accessible by table, primary key, and column.
4. Null values must be uniformly treated as “missing information” not as empty strings, blanks, or zeros.
5. Metadata (data about the database) must be stored in the database just as regular data is.
6. A single language must be able to define data, views, integrity constraints, authorization, transactions, and data manipulation.
7. Views must show the updates of their base tables and vice versa.
8. A single operation must be able to retrieve, insert, update, or delete data.
9. Batch and end-user operations are logically separate from physical storage and access methods.
10. Batch and end-user operations can change the database schema without having to recreate it or applications built upon it.
11. Integrity constraints must be available and stored in the metadata, not in an application program.
12. The data manipulation language of the relational system should not care where or how the physical data is distributed and should not require alteration if the physical data is centralized or distributed.
13. Any row-processing done in the system must obey the same integrity rules and constraints that set-processing operations do.

3. What Can SQL do?

SQL can execute queries, retrieve data, insert, update, delete records, create new databases, create new tables and views in a database, can set permissions on tables, procedures, and views. Can create stored procedures in a database

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

The process of separating data into distinct, unique sets is called normalization.

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

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

Yes. The exact syntax is not important.

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

select \*; Select \* from checks Select amount name payee FROM checks;

The correct syntax is SELECT FROM <COLUMN NAMES> SEPARATED BY COMMA;

7. Which of the following SQL statements will work?

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

select \* from checks;

select \* from checks /

8 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.
2. Rewrite the query from exercise 1 so that the topCharge will appear as the first column in your query results.
3. Using the arrests table, write a query to return all the unique topCharges.

1 SELECT officerId, topCharge FROM arrests;

2 SELECT topCharge, officerId FROM arrests;

3 SELECT DISTINCT topCharge FROM arrests;

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.
2. Write a query that returns everyone who lives in Illinois with a first name of AL.
3. What shorthand could you use instead of WHERE a >= 10 AND a <=30?
4. What will this query return?

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

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

NAME ST

AL FROM IL

1. SELECT \* FROM doubleAgents WHERE Lastname LIKE 'M%';
2. SELECT \* FROM doubleAgents where STATE='IL' AND FIRSTNAME = 'AL';
3. BETWEEN 10 AND 30;
4. NOTHING
5. SELECT (FIRSTNAME || ‘FROM’) NAME, STATE FROM DOUBLEAGENTS WHERE LASTNAME =’BUNDY’ AND STATE=’IL’;
6. 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 LASTNAME || ‘,’ || FIRSTNAME NAME,

AREACODE || ‘-‘ || PHONE PHONE FROM DOUBLEAGENTS WHERE

AREACODE BETWEEN 300 AND 345;

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

INITCAP converts the first letter of each word to a capital letter

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

Group and aggregate functions.

1. Will this query work?

SELECT COUNT(LASTNAME) FROM CHARACTERS;

Yes.

1. How about this one?

SELECT SUM(LASTNAME) FROM CHARACTERS

No.

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

CONCAT and ||, also

SELECT FIRSTNAME+’, ‘+LASTNAME AS FULLNAME FROM TABLENAME;

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

SELECT COUNT(\*) FROM drone\_strikes;

37 records in drone\_strikes table

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

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

Should be SELECT SUBSTR (LASTNAME, 1, 5) FROM NAME\_TBL;

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.

SELECT \* FROM SHOOTSTATS WHERE (HITS/SHOTSTAKEN)<.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 |

SELECT SUBSTR (FIRST, 1,1) ||’.’||

SUBSTR (MIDDLE, 1,1) ||’.’||

SUBSTR (LAST, 1,1) ||’.’|| INITIALS, BADGEID CODE FROM OFFICERS WHERE BADGEID=32

1. 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 GROUP BY groups data sets that have been manipulated by various functions. The GROUP BY acts like the ORDER BY -- it orders the results of the query in the order the columns are listed in the GROUP BY.

1. Will this SELECT work?

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

NO.

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;

IT WILL RETURN NUMBER OF PEOPLE ON EACH TEAM WITH SICKLEAVE >=30.

WHAT IS THE QUESTION?