

CPT103-Tutorial

Yu LIU (Yu.Liu02@xjtu.edu.cn)
Department of Computing

Week 11, Dec 2023



- Interactive Tutorial : 85 min
- Q&A : 15 min



Instructions

SQL stands for _____

grade

structured query language

structured query language

Structured query
language

Structure Query Language

Structured Query
Language

standard query language

Structured query
language

SQL stands for _____

Structured query
language

Structured Query
Language

Structured Query
Language

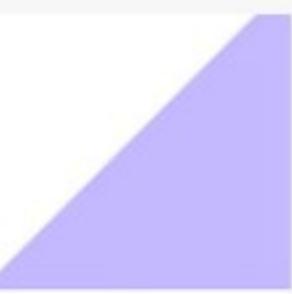
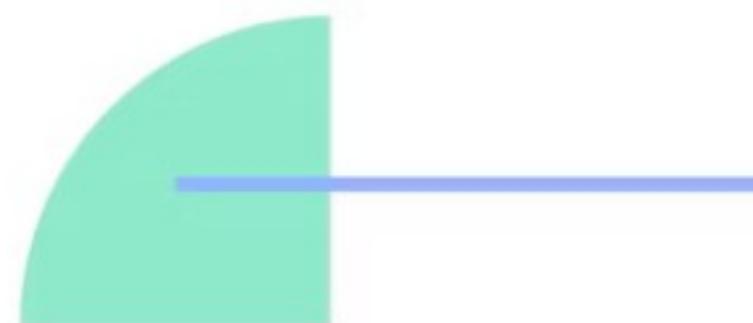
sequential query language

a programming language

subquerylanguage

Structural query language

Structured query
language



SQL stands for _____

a kind of database

CPT103

Structured Query
Language

structured query language

structured querylanguage

Structured Query
Language

structure query language

structured query language

SQL stands for _____

Structured Query
Language

standard query language

Structured Query
Language

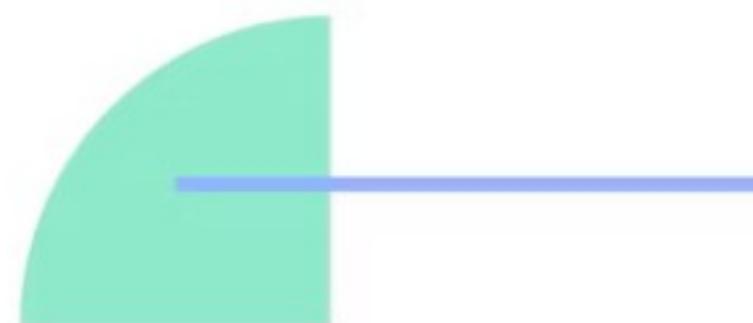
Structured Query
Language

structured query language

Database which stored
data

structured query language

Structured Query
Language



SQL stands for _____

structured query language

Structure Query Language

Structure query language

Structured Query
Language

STRUCTURE QUERY
LANGUAGE

structured query language

Structured query
language

Structured Query
Language

SQL stands for _____

A database language

Structured Query
Language

Structured Query
Language.

structured query language

Structure query language

Structure query language

SQL

structure query language.

SQL stands for _____

structured query language

Structure query language

Structure Query Language

structured query language

structure query language

structured quary language

Structured Query
Language

Structured Query
Language

SQL stands for _____

structured query language

score

structure query language

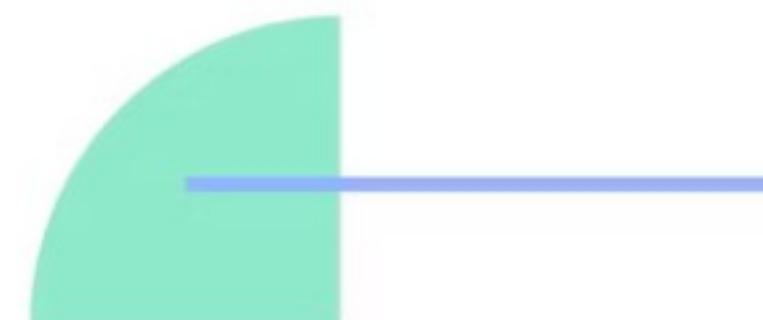
Structured query
language

Structured Query
Language

structure querly language

Structured Query
Language

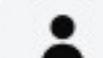
structured query language



45



115



SQL stands for _____

Structure Query Language

Structured query
Language

Structured Query
Language

Structured Query
Language

structred query language

structured query language

structural query language

structured query language

SQL stands for _____

Structure query language

structure query language

Structured Query
Language

Structured querry
language

programlanguage

structured query language

Structured query
language

Structured query
language

SQL stands for _____

Structured query
language

Structured Query
Language

structured query language

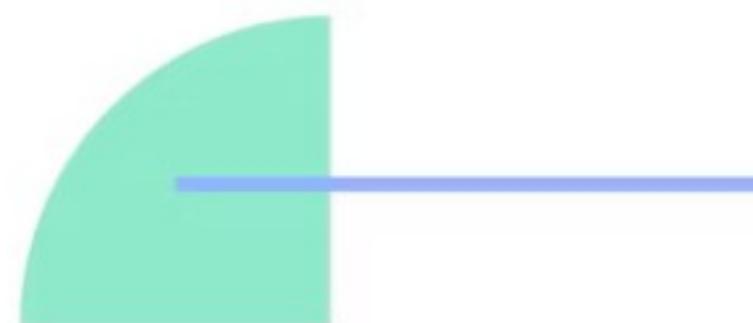
structured query language

structured query language

Structured query
language

Structure Query Language

structured query language



SQL stands for _____

structure query language

Structured query
language

structured query language

Structured Query
Language

Structural query language

structured query language

Stuctured query language

Structured query
language

SQL stands for _____

StructuredQueryLanguag
e

Structured query
language

structured query language

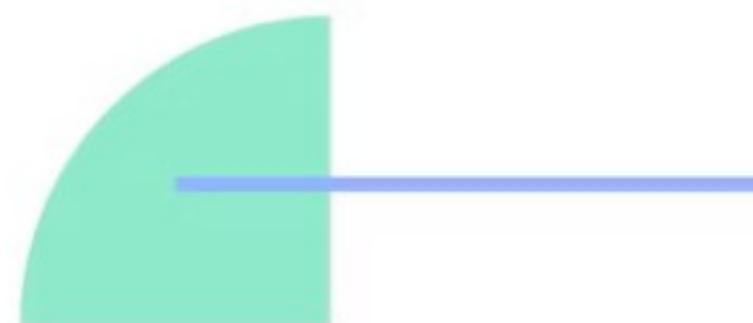
Structural query language

structure query language

Structured query
language

Structured Query
Language

Structured Query
Language



SQL stands for _____

structured query language

structured query language

Structured Query Language. It is a standardized programming language used for managing databases

structured query language

SQL stands for _____

The SQL acronym stands for Structured Query Language. It is a universal programming language used to manage relational databases (RDBMS)

Structured query language

structured query language

SQL, DDL, DML

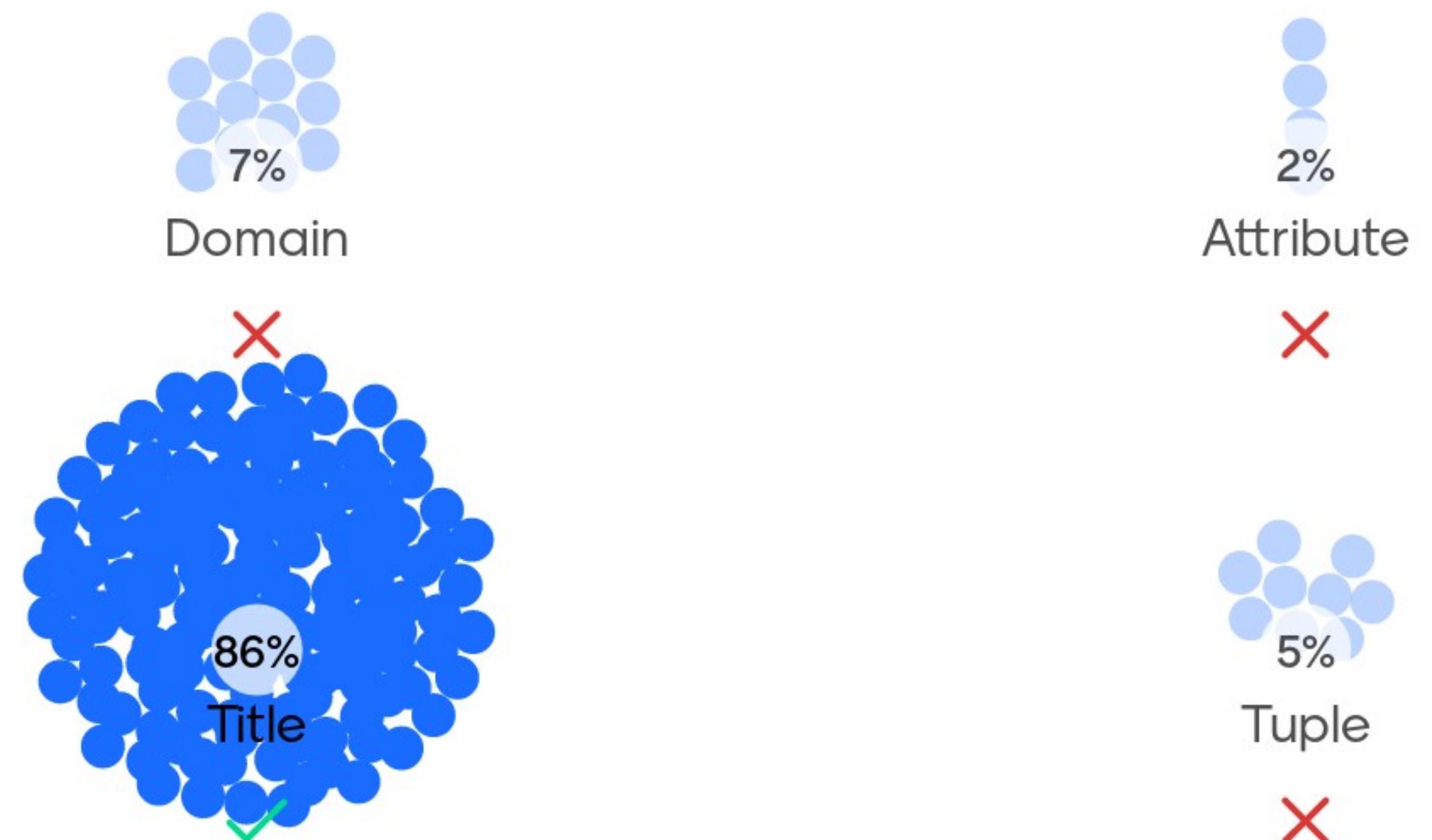
SQL is Structured Query Language.

DDL is Data Definition language, dealing with database schemas and descriptions, of how the data should reside in the database.

DML is Data Manipulation Language, dealing with data manipulation, and includes most common SQL statements.

DDL	DML
CREATE ALTER DROP ...	SELECT INSERT UPDATE DELETE ...

Which concept below is NOT in the relational model?



Relational Model

- It represents the method of structuring data using relations, which are essentially tables.
- A relation is a table with columns and rows.
- An attribute is a named column of a relation; A functional dependency is a relationship between or among attributes.
- A tuple is a row of a relation.
- A domain is the set of allowable values for one or more attributes.
- The cardinality of a relation is the number of tuples it contains.

Total # of rows is **Cardinality**

Row or **Tuple**

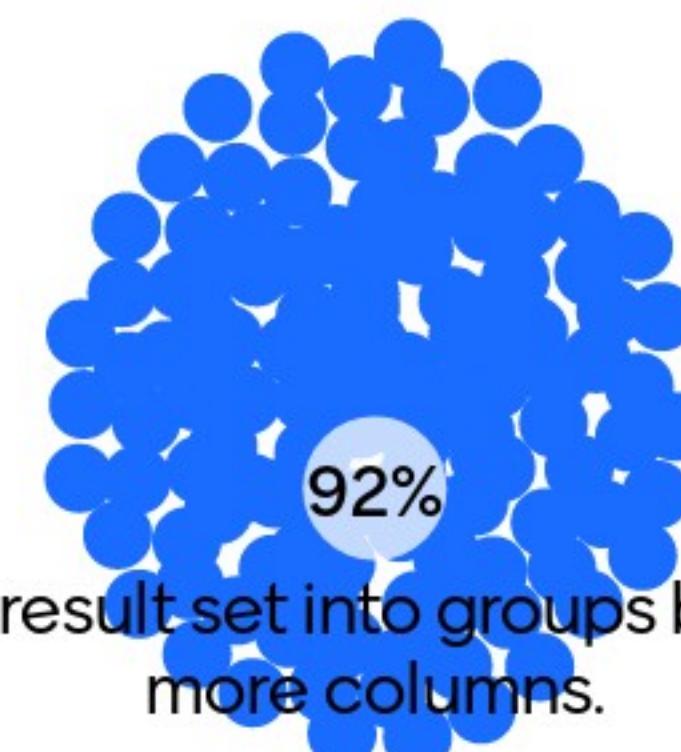
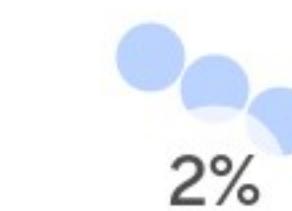
Total # of column is **Degree**
Column or **Attributes**

Table also called **Relation**

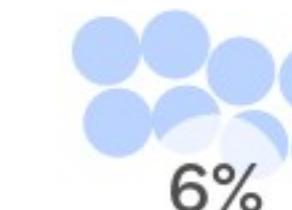
StudentID	StudentName	Programme
1719442	Paul	CS
1877663	Eric	CS
1988342	Thomas	CS

What is the purpose of the SQL keyword 'GROUP BY' in a SELECT statement?

- a. Joins tables based on a related column between them.
- b. Divides the result set into groups based on one or more columns.



- c. Sorts the result set in descending order.



- d. Updates data in a database table.



0%

Get the total number of books sold per genre

```
SELECT genre, SUM(books_sold) AS total_books_sold  
FROM sales  
GROUP BY genre;
```

In this query:

- SELECT is used to specify the columns we want to retrieve, here genre and the sum of books_sold.
- SUM(books_sold) calculates the total number of books sold for each genre.
- AS total_books_sold gives a name (alias) to our calculated sum for easier reference.
- FROM sales specifies the table from which to retrieve the data.
- GROUP BY genre groups the results by the genre column, so the sum of books_sold is calculated for each genre separately.

Some common SQL keywords and their primary functions

- HAVING: Applies a condition to filter groups created by the GROUP BY clause.
- ORDER BY: Specifies the order in which to return the rows, either ascending (ASC) or descending (DESC).
- DISTINCT: Removes duplicate rows from the result set.
- LIMIT: Specifies the maximum number of records to return (useful in large databases).
- LIKE: Used in a WHERE clause to search for a specified pattern in a column.
- IN: Allows you to specify multiple values in a WHERE clause.

- HAVING

Applies a condition to filter groups created by the GROUP BY clause.

This query selects the total salary for each department from the employees table, but only includes those departments where the total salary exceeds 100,000.

```
Run SQL query/queries on database yuliu:   
1 SELECT department, SUM(salary)  
2 FROM employees  
3 GROUP BY department  
4 HAVING SUM(salary) > 100000;  
5
```

- ORDER BY

Specifies the order in which to return the rows.

This query selects names and ages from the people table and orders the results in descending order of age.

Run SQL query/queries on database yuliu: 

```
1 SELECT name, age  
2 FROM people  
3 ORDER BY age DESC;  
4
```

- DISTINCT

Removes duplicate rows from the result set.

This query selects unique country names from the cities table, removing any duplicates.

```
Run SQL query/queries on database yuliu:   
1 SELECT DISTINCT country  
2 FROM cities;  
3
```

- LIMIT

Specifies the maximum number of records to return.

This query selects all columns from the products table but only returns the top 10 most expensive products.

Run SQL query/queries on database yuliu: 

```
1 SELECT *  
2 FROM products  
3 ORDER BY price DESC  
4 LIMIT 10;  
5
```

- LIKE

Used in a WHERE clause to search for a specified pattern in a column.

This query selects all columns from the customers table where the customer's name starts with the letter 'J'.

```
Run SQL query/queries on database yuliu: ⚙  
1 SELECT *  
2 FROM customers  
3 WHERE name LIKE 'J%';  
4  
5  
6
```

-IN

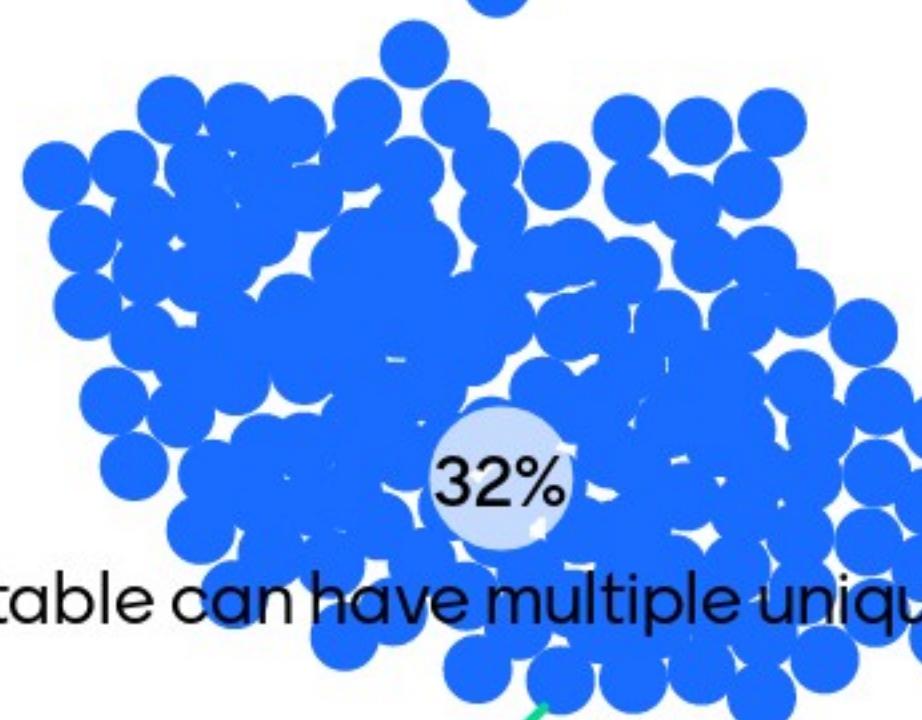
Allows you to specify multiple values in a WHERE clause.

This query selects all columns from the movies table where the genre is either 'comedy', 'drama', or 'action'.

```
Run SQL query/queries on database yuliu:   
1 SELECT *  
2 FROM movies  
3 WHERE genre IN ('comedy', 'drama', 'action');
```

Which of the following are true about primary and unique keys? (Multiple choice)

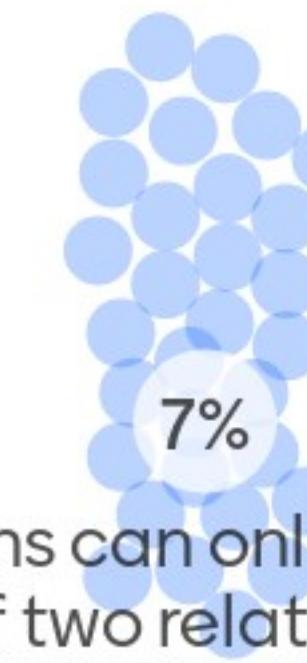
a. A table can have multiple unique keys.



32%



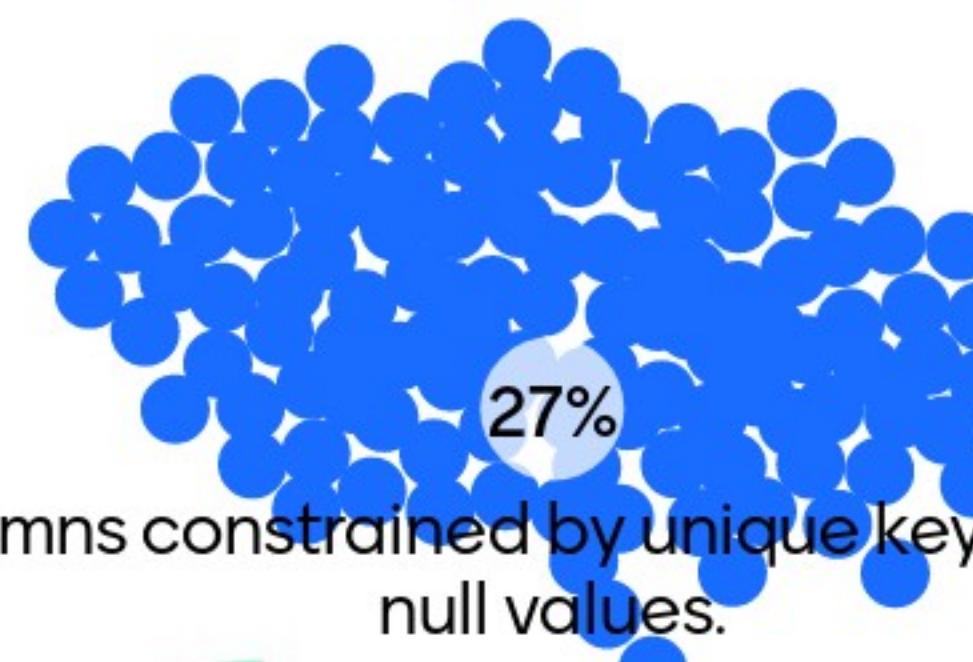
b. UNION operations can only be applied when the primary keys of two relations are the same.



7%



d. Columns constrained by unique keys can have null values.



27%



e. The column definition "studentID int primary key default null" will not cause any errors.



c. Columns constrained by primary keys can have null values.



1%



f. Columns with primary key constraints can be super keys.



30%



Primary Key	Unique Key
Unique identifier for rows of a table	Unique identifier for rows of a table when primary key is not present
Cannot be NULL	Can be NULL
Only one primary key can be present in a table	Multiple Unique Keys can be present in a table

Primary Key, Unique Key

Union

- In SQL, the UNION operator is used to combine the result sets of two or more SELECT statements.
- It removes duplicate rows between the various SELECT statements.
- Each SELECT statement within the UNION must have the same number of columns in the result sets with similar data types.

grade10_students

student_id	student_name
1	Alice
2	Bob
3	Charlie

grade11_students

student_id	student_name
4	David
5	Emma
2	Bob

Example

Using UNION:

```
SELECT student_id, student_name FROM  
grade10_students
```

```
UNION
```

```
SELECT student_id, student_name FROM  
grade11_students;
```

student_id	student_name
1	Alice
2	Bob
3	Charlie
4	David
5	Emma

Super Key

- Uniqueness: The most fundamental characteristic of a super key is that it must uniquely identify each row in a table. This means no two rows can have
- Can Be a Combination of Columns: A super key can consist of a single column or a combination of multiple columns.
- May Contain Additional Attributes: A super key can contain additional attributes that are not necessary for unique identification.
- Every Table has at Least One Super Key: This can be a single attribute or a combination of all the attributes in the table.
- Relationship to Primary Key: Every primary key is a super key, but not every super key is a primary key.

Given the CREATE TABLE statement, which of the following is/are correct?

a. It tries to create a table with 2 primary keys.



d. 2 unique keys will be created.



b. It tries to create a table with 1 primary key consisting of 2 columns.



e. 3 unique keys will be created.



41%

11%

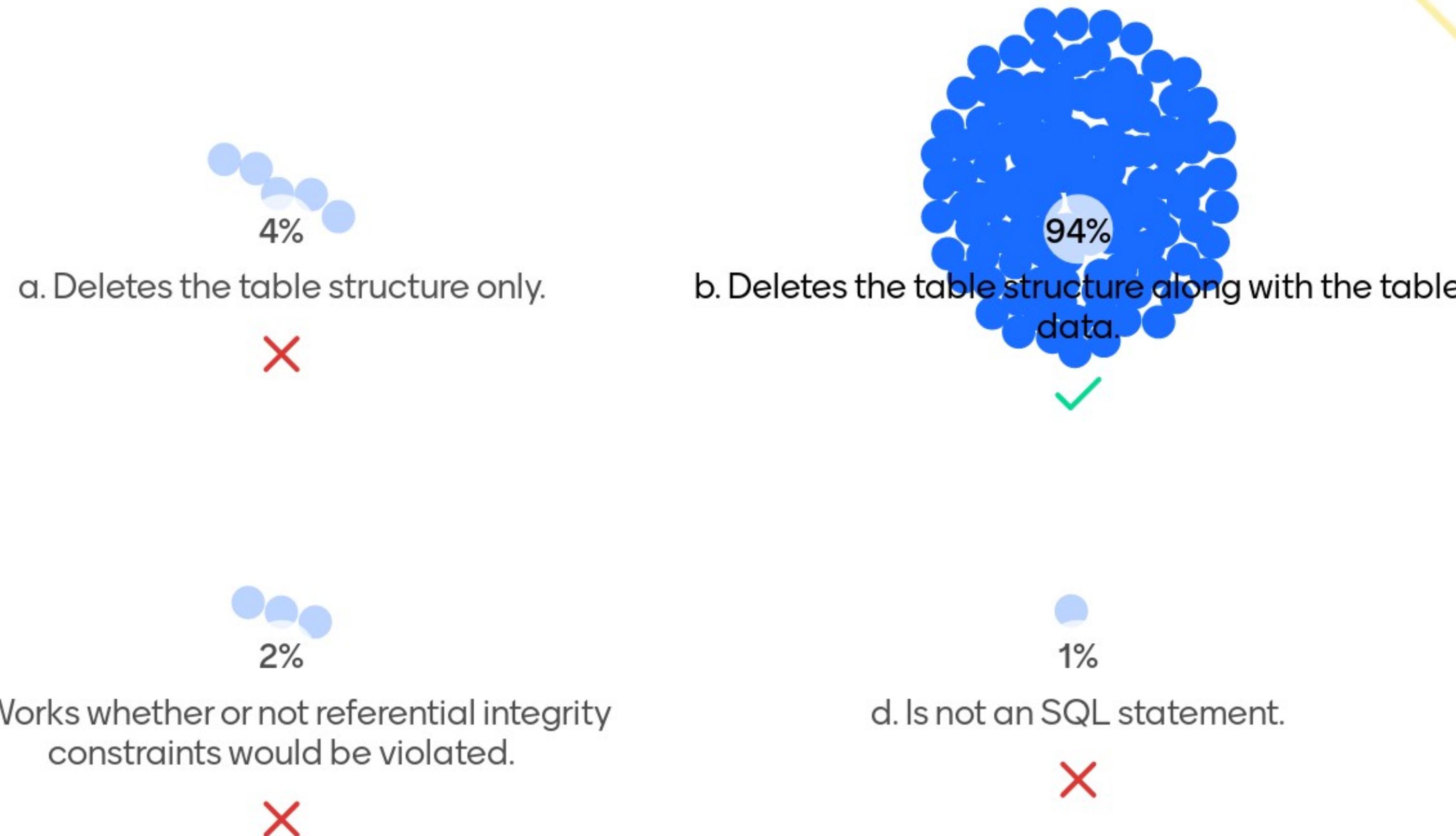
23%

16%

9%

```
create table A (
    col1 int primary key,
    col2 int primary key,
    col3 int unique key,
    col4 int,
    col5 int,
    constraint cons1 UNIQUE (col4, col5)
);
```

The DROP TABLE statement:



Which of the following are true about applying a foreign key constraint? (Assume we use the default settings for MySQL)

The number of attributes involved with the foreign key in the referencing table should be the same as the number of attributes in the referenced table.

18%

A table can have foreign keys referencing itself.

13%

The column(s) involved with the foreign key in the referencing table must be a primary/unique key.

11%

The attributes(s) involved with the foreign key in the referenced table can be a subset of the attributes in the primary key.

18%

A table can have multiple foreign keys.

22%

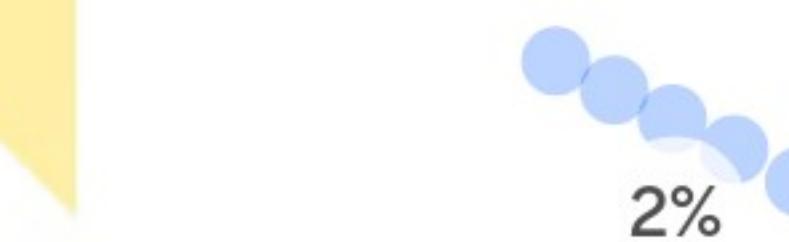
The foreign key can still be applied when existing data violates the referential integrity involved in this foreign key.

2%

A column that references other columns can have null values.

17%

Which syntax below is/are able to change the datatype of an existing column in MySQL?



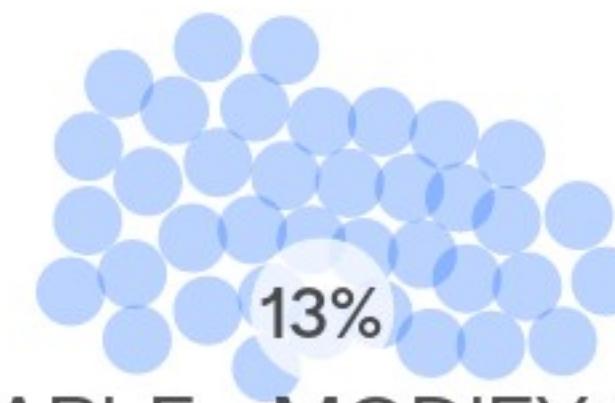
ALTER TABLE ... ALTER COLUMN ...



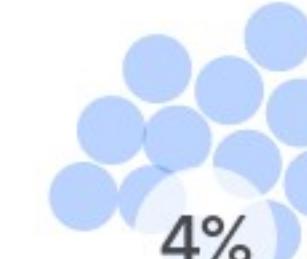
ALTER TABLE ... UPDATE COLUMN ...



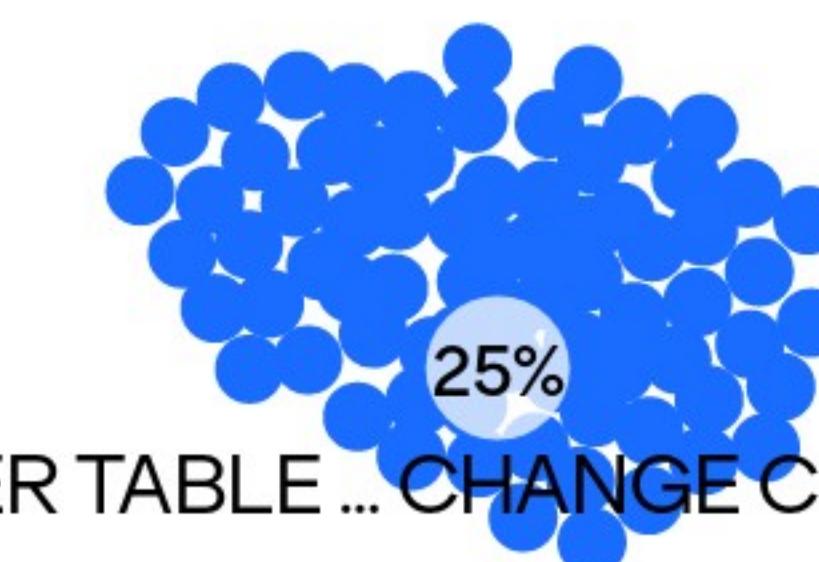
ALTER TABLE ... MODIFY COLUMN ...



UPDATE TABLE ... MODIFY COLUMN ...



UPDATE TABLE ... CHANGE COLUMN ...



ALTER TABLE ... CHANGE COLUMN ...



MODIFY, CHANGE

- MODIFY COLUMN command is used to change the data type of a column in a table.
- ALTER TABLE CHANGE COLUMN statement is used to change the data type and name of a column.
- ALTER TABLE employees MODIFY COLUMN age VARCHAR(3);
- ALTER TABLE employees CHANGE COLUMN name employee_name VARCHAR(100);

employees

Column Name	Data Type
employee_id	INT
name	VARCHAR
age	INT

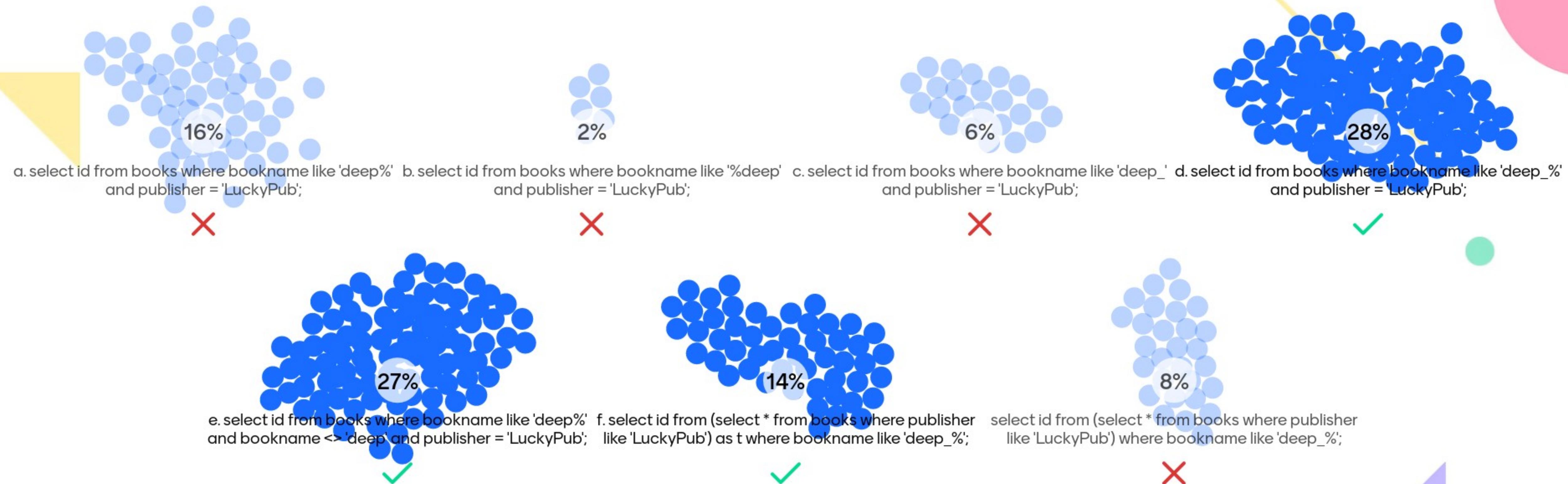
Select name from staff where name <> 'Annie' and name != 'Brown'.
The query above will return ___ rows.

129 responses

321
540
three
idk
345

	staff_id	staff_name	staff.age
1	1	Annie	28
2	2	Brown	31
3	3	George	26
4	4	Kathy	21
5	5	Annie	21

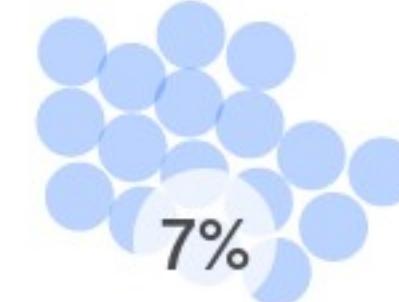
Which of the following able to find all the IDs of books whose names start with "Deep" but cannot just be "Deep", and the publisher is "LuckyPub":



Wildcards: %, _

- The percent sign (%) represents zero, one, or multiple characters.
- The underscore sign (_) represents one, single character.

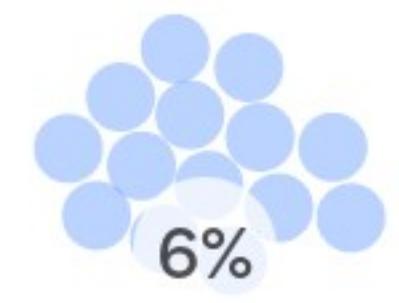
Which of the following statements are able to find persons who earn the most?



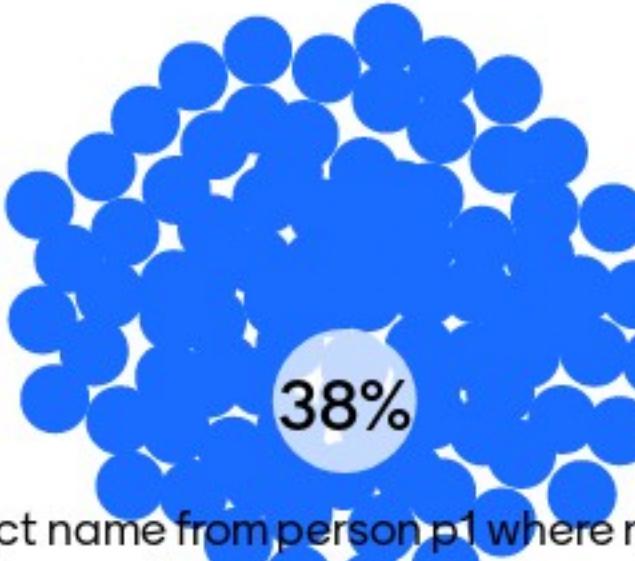
a. select name from person where income >= any
(select income from person);



b. select name from person where income >= all
(select income from person);



c. select name from person p1 where exists (select
income from person p2 where p1.income <=
p2.income);



d. select name from person p1 where not exists
(select income from person p2 where p1.income <
p2.income);



	name	income	married
1	Mirana	1,500	1
2	Giovanni	1,450	0
3	Kerry	1,700	0
4	Fiona	2,000	1
5	Morris	2,000	1

ALL, ANY

- If the subquery is preceded by the keyword ALL, the condition will be true only if it is satisfied by all values produced by the subquery.
- If the subquery is preceded by the keyword Any, the condition will be true only if it is satisfied by one or more values produced by the subquery.

Data in third normal form (3NF) contains which of the following?

a. partial dependencies



c. repeating groups



4

b. transitive dependencies



21

d. none of the above



2

99

Partial Dependency

A functional dependency $A \rightarrow B$ is a full functional dependency if removal of any attribute from A results in the dependency no longer existing. A functional dependency $A \rightarrow B$ is a partially dependency if there is some attribute that can be removed from A and yet the dependency still holds.

FD : $AB \rightarrow CDE$ where AB is Primary Key.

Then, { $A \rightarrow C; A \rightarrow D; A \rightarrow E; B \rightarrow C; B \rightarrow D; B \rightarrow E$ } all are Partial Dependencies.

Partial Dependency occurs when there is PK consists of many keys (composite key) and non-primary keys depended on part of this composite key.

Example 1: Suppose you have the following table:

Both (stud_id, course_name) are PK, and m_grade and f_grade depend on this PK.

So if m_grade and f_grade are partially depended on only course_name instead on all PK components, the table will not be in the 3nf.

stud_id	course_name	m_grade	f_grade
111	Programming	44	79
112	DB	50	88
111	OOP	60	80

Transitive Dependency

A condition where A, B, and C are attributes of a relation such that if $A \rightarrow B$ and $B \rightarrow C$, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

Non-attribute key determines non-attributes key.

Example 2: The table below:

The functional dependency $\{Book\} \rightarrow \{Author Nationality\}$ applies; that is, if we know the book, we know the author's nationality. Furthermore:

$\{Book\} \rightarrow \{Author\}$

$\{Author\}$ does not $\rightarrow \{Book\}$

$\{Author\} \rightarrow \{Author Nationality\}$

Therefore $\{Book\} \rightarrow \{Author Nationality\}$ is a transitive dependency.

Book	Genre	Author	Author Nationality
<i>Twenty Thousand Leagues Under the Sea</i>	Science Fiction	Jules Verne	French
<i>Journey to the Center of the Earth</i>	Science Fiction	Jules Verne	French
<i>Leaves of Grass</i>	Poetry	Walt Whitman	American
<i>Anna Karenina</i>	Literary Fiction	Leo Tolstoy	Russian
<i>A Confession</i>	Religious Autobiography	Leo Tolstoy	Russian

Table employees

	emp_id	emp_name	job_name	hire_date	salary	dep_id
1	Clare	President		2001-01-15	8000.00	1001
2	Bob	Manager		2001-03-12	5600.00	2001
3	Frank	Manager		2001-05-16	5575.00	1001
4	John	Manager		2001-01-18	5720.00	3001
5	Blaze	Scientist		2002-10-15	4050.00	2001
6	Andy	Scientist		2002-08-12	5000.00	2001
7	Tom	Analyst		2003-02-20	4075.00	2001
8	Jason	Accountant		2001-07-18	3500.00	1001
9	Andrew	Salesman		2005-06-15	3100.00	1001
10	Mason	Salesman		2006-03-05	3200.00	3001

Table department

dep_id
1001
2001
3001
4001

location_id
2500
2100
3200
1600

Table Location

location_id	address	postcode	city
1600	London Road	L1 0L7	Liverpool
2000	Eiffel Avenue	P7 005	Paris
2100	Magdalen Centre	OX7 2BD	Oxford
2500	Bridge Road	SW5 3BZ	London
3000	The fifth Avenue	NY7 881	New York
3200	Oxford Street	MA5 X79	Manchester
3500	Colosseum Piazza	RM7 999	Rome

Which one(s) of the following queries returns the list of employees whose salaries are more than 5500?

2%

```
SELECT emp_name, salary > 5500 FROM  
employees;
```



70%

```
SELECT emp_name, salary FROM employees  
WHERE salary > 5500;
```



23%

```
SELECT emp_name, salary FROM employees  
HAVING salary > 5500;
```



5%

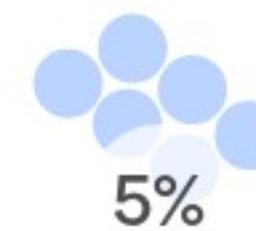
```
SELECT emp_name, salary FROM employees  
ORDER BY salary > 5500;
```



WHERE, HAVING

- HAVING Clause: The HAVING clause is used to filter groups after the GROUP BY clause has aggregated the data.; It works on aggregated data.
- The WHERE clause is used to filter rows before any groupings are made.
- `SELECT emp_name, salary FROM employees WHERE salary > AVG(salary);`
- `SELECT emp_name, salary FROM employees HAVING salary > AVG(salary);`

Which query below is able to find the name and salary of the employees whose salary is higher than the average salary of all departments?



SELECT * FROM employees WHERE salary > ANY(SELECT avg(salary) FROM employees GROUP BY Dep_id);
SELECT * FROM employees WHERE salary > ANY(SELECT avg(salary) FROM employees HAVING Dep_id);



SELECT * FROM employees WHERE salary > ANY(SELECT avg(salary) FROM employees GROUP BY Dep_id);
FROM employees FROM employees HAVING Dep_id);



SELECT * FROM employees WHERE salary > ALL(SELECT avg(salary) FROM employees HAVING Dep_id);
SELECT * FROM employees WHERE salary > ALL(SELECT avg(salary) FROM employees GROUP BY Dep_id);



SELECT * FROM employees WHERE salary > ALL(SELECT avg(salary) FROM employees GROUP BY Dep_id);
FROM employees FROM employees HAVING Dep_id);



R		
A	B	C
7	1	5
5	2	1
4	6	1
7	3	1

S		
A	B	C
3	5	1
6	3	6
8	7	3
4	2	3

Which of the following statements can return more than 1 row of results?



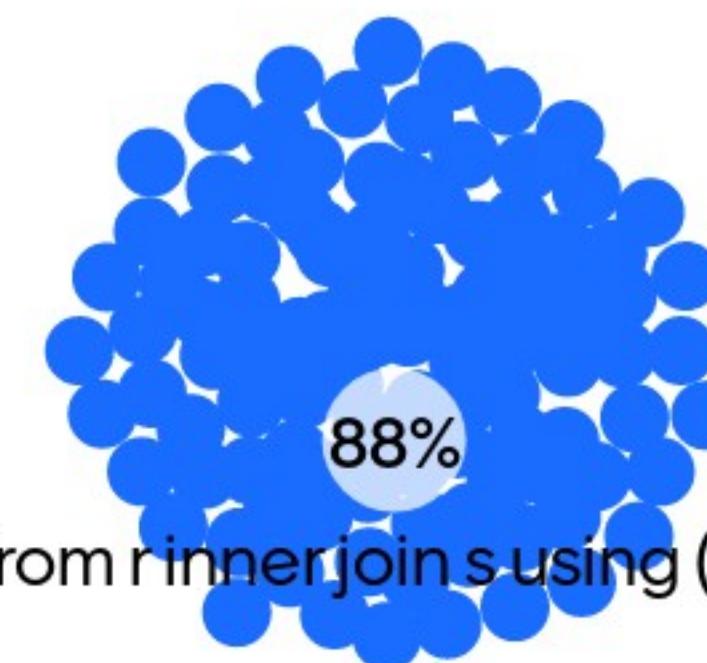
a. Select * from r, s where a > 7;



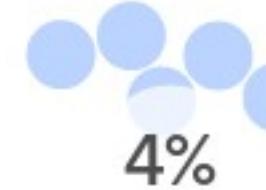
b. Select * from r where b > a;



c. Select * from s where c = b;



d. Select * from r inner join s using (b) where b > 1;



e. Select * from r inner join s using (a) where a < 9;



Inner join, left join, right join

- Select * from r left join s on r.b > s.b;
- Select * from r right join s on r.b > s.b;

Select * from r left join s on r.b > s.b;

性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]

显示全部 行数: 25 过滤行: 在表中搜索

额外选项

A	B	C	A	B	C
4	6	1	3	5	1
4	6	1	6	3	6
4	6	1	4	2	3
7	3	1	4	2	3
7	1	5	NULL	NULL	NULL
5	2	1	NULL	NULL	NULL

Select * from r right join s on r.b > s.b;

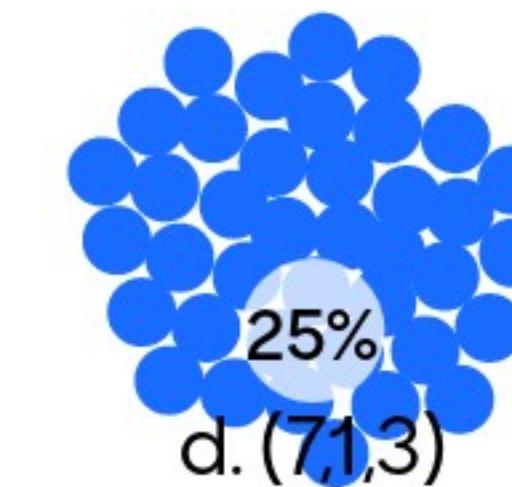
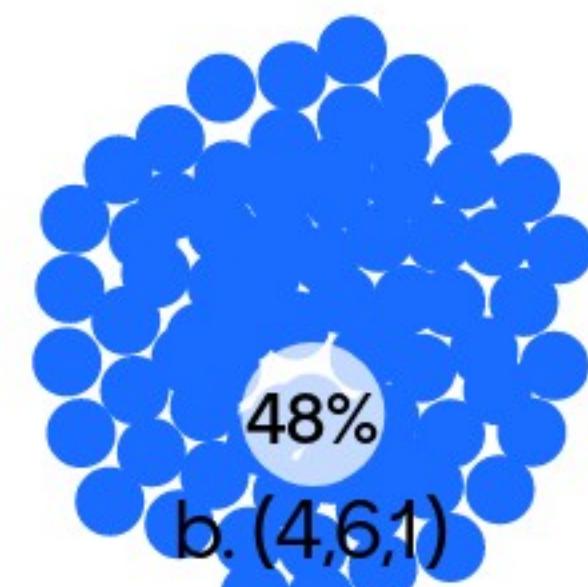
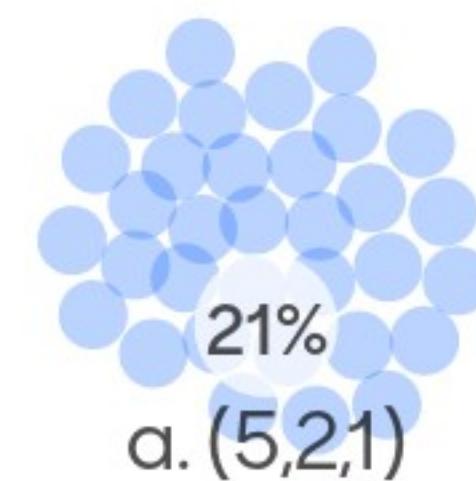
性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]

显示全部 行数: 25 过滤行: 在表中搜索

额外选项

A	B	C	A	B	C
4	6	1	3	5	1
4	6	1	6	3	6
4	6	1	4	2	3
7	3	1	4	2	3
NULL	NULL	NULL	8	7	3

After executing: select * from r where a in (select distinct b from s); Which of the following rows are NOT in the result?



Any Questions?

See you next week 

outer join handles nulls

quiz on Saturday?

When will the test be

can u show this ppt on
LMO, thanks

