SQL FUNDEMENTALS

1. DATABASES

What is a database?

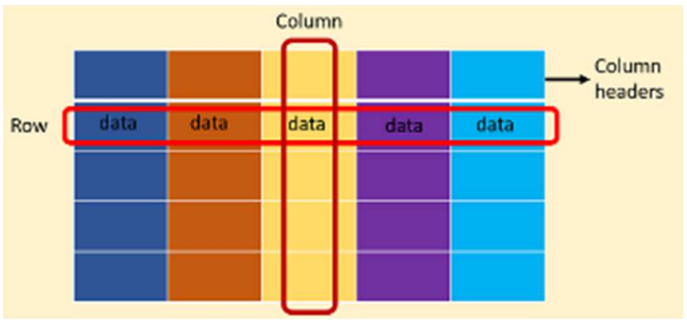
Database is a collection of data stored in a computer system. People use data and databases with or without awareness in their daily life activities. All use databases to keep track of your data.

Websites = store "accessing date and time, your location, your browser info"
Social media platforms = use databases to store data like members, their friends, member activities, messages, advertisements, etc.

What is in a database?

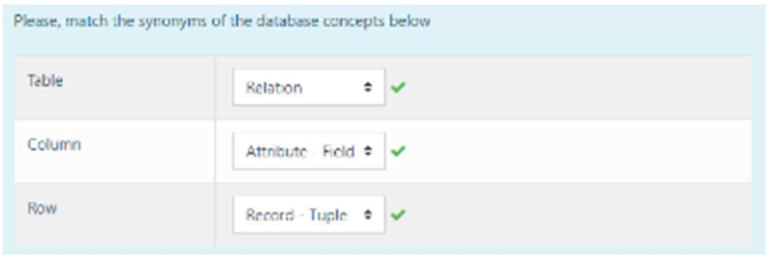
The information inside the database is grouped into tables. **Structured data** here means table. A table is made up of columns and rows. Columns and rows together make up a table.

Row = Single set of columns / "a record" or a "tuple"
Column = " a field" or "attribute"
Table = "a relation"



Interview Q&A:
Q: What is a table, column and row?
A: A table is an organized collection of data stored in the form of columns and rows. Columns are vertical and rows as horizontal. Columns are called fields while and rows are called as records.

In-class Practise:



Please, match the synonyms of the database concepts below

Table	Relation ✓
Column	Attribute - Field ✓
Row	Record - Tuple ✓

Types of Databases

There are two main database storage types:

"Relational Database - SQL"   &   "Non-Relational Database – NoSQL"

Relational database stores and provides access to data points that are related to one another. Each row is "a record" with a unique ID called the *key*. And each record usually has a value for each attribute. Relational database term invented by E. F. Codd at IBM in 1970.

RDBMS: “Relational Database Management System” A software system - used to maintain relational databases. SQL: “Structured Query Language” - accepted as the standard RDBMS language. So we usually prefer to call Relational Database as SQL and Non-Relational database as NoSQL. Ide is SQLite.

Structured Query Language (SQL)

Writing SQL declaration ⇨ hand it to the DBMS ⇨ DBMS executes internal code

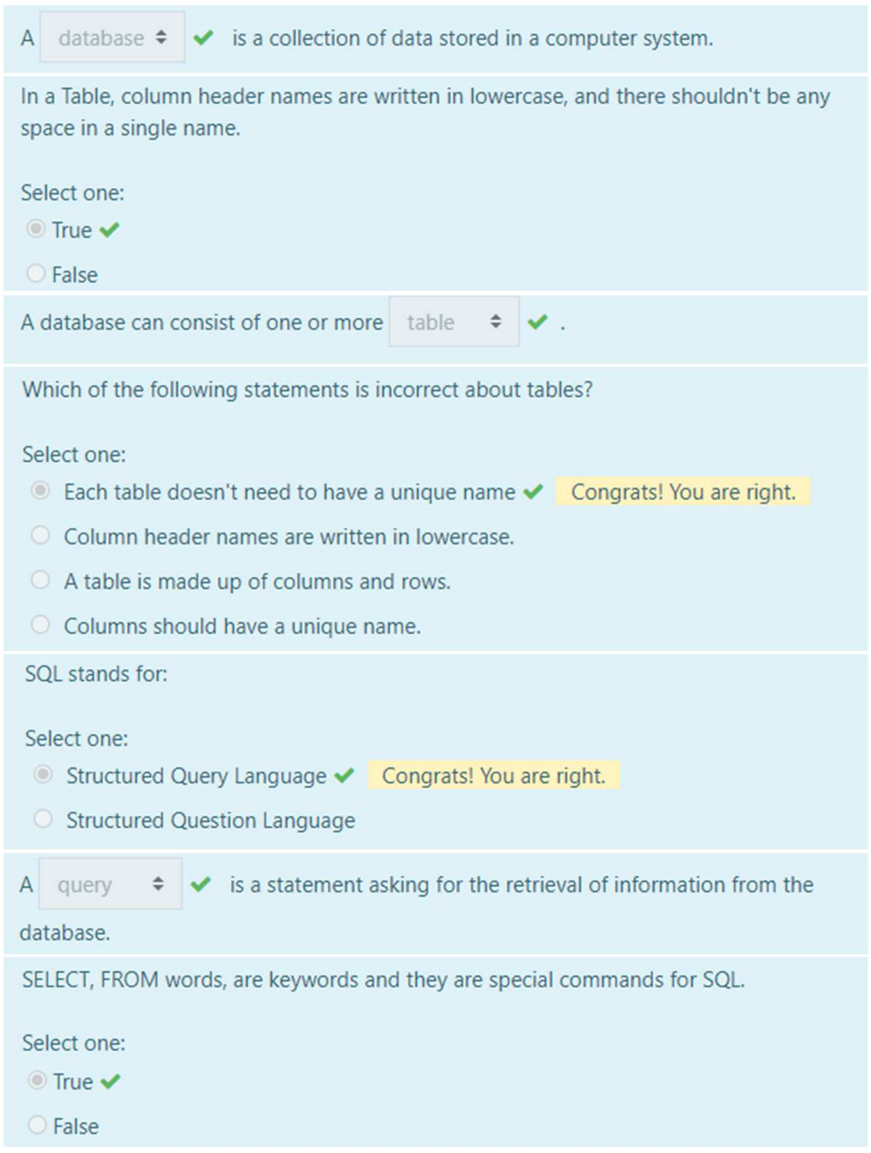
SQL is not a programming language, but it’ a query language. But somehow with appends SQL can have looping and control structures.

SQL Language Elements

It is easy to understand SQL language structure. Cause it looks like plain English. Let’s check following ex.;

SELECT first_name FROM employees; ⇨ it’s called **statement.**
SELECT and FROM ⇨ words are **keywords.**
first_name and employees ⇨ are **identifiers.**
There are other SQL elements which we will cover later.

Check Yourself Questions:



A database ✓ is a collection of data stored in a computer system.

In a Table, column header names are written in lowercase, and there shouldn't be any space in a single name.

Select one:
[✓ True] ✓
False

A database can consist of one or more table ✓ .

Which of the following statements is incorrect about tables?

Select one:
[✓ Each table doesn't need to have a unique name] ✓ Congrats! You are right.
Column header names are written in lowercase.
A table is made up of columns and rows.
Columns should have a unique name.

SQL stands for:

Select one:
[✓ Structured Query Language] ✓ Congrats! You are right.
Structured Question Language

A query ✓ is a statement asking for the retrieval of information from the database.

SELECT, FROM words, are keywords and they are special commands for SQL.

Select one:
[✓ True] ✓
False

2. SELECT Statement

Introduction & Basic Syntax

⇨ The SELECT statement is used to select data from a database.
⇨ SELECT statement is used with FROM keyword.
Syntax; ⇨ SELECT column_name(s) FROM table_name;

SQL statements start with a keyword like SELECT, INSERT, UPDATE, DELETE
Writing SQL commands in the upper-case is the most common and preferred style.
Spaces and empty lines are ignored in SQL.

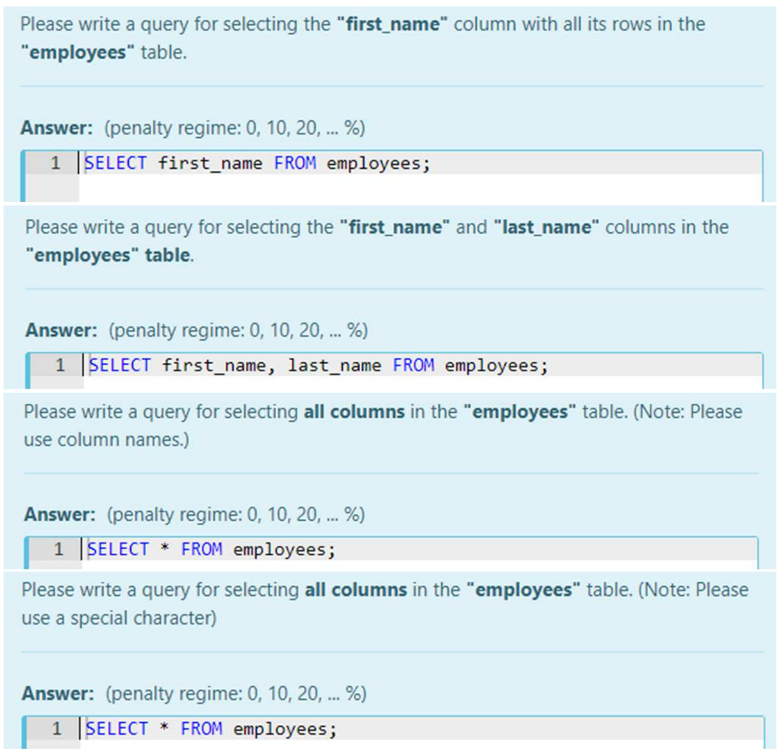
Selecting Multiple Columns

Syntax; ⇨ SELECT column1, column2 FROM table1;

Selecting ALL Columns

Syntax; ⇨ SELECT * FROM table1; --Asterisk

Check Yourself Questions:



Please write a query for selecting the **"first_name"** column with all its rows in the **"employees"** table.

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT first_name FROM employees;
```

Please write a query for selecting the **"first_name"** and **"last_name"** columns in the **"employees" table**.

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT first_name, last_name FROM employees;
```

Please write a query for selecting **all columns** in the **"employees"** table. (Note: Please use column names.)

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT * FROM employees;
```

Please write a query for selecting **all columns** in the **"employees"** table. (Note: Please use a special character)

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT * FROM employees;
```

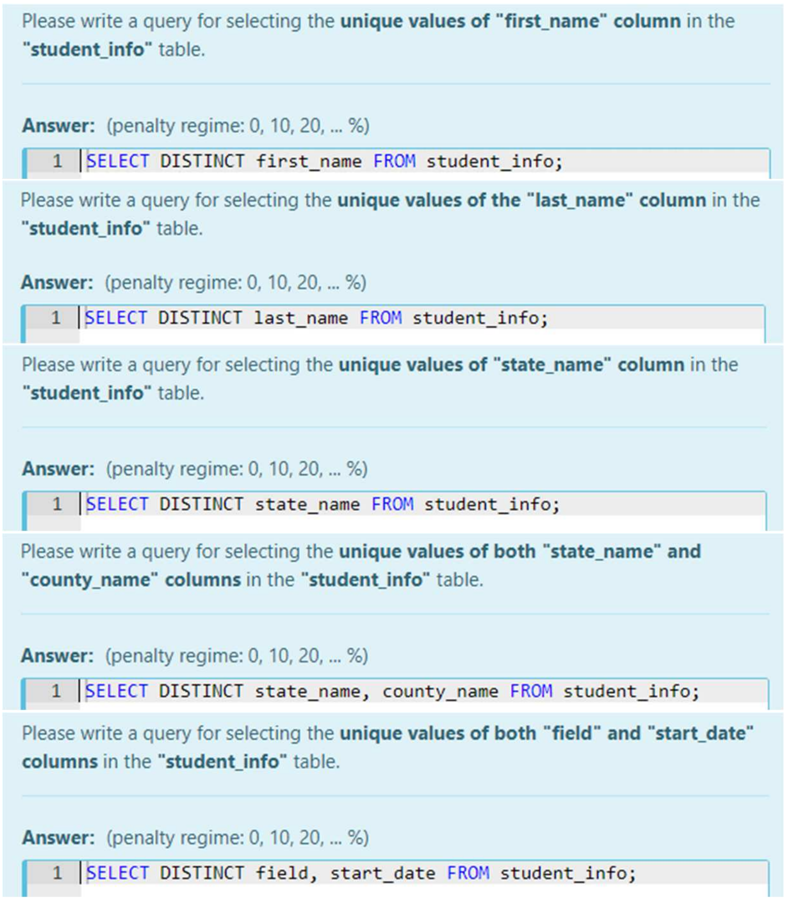
3. DISTINCT Clause

Introduction

When need the distinct duplicate values as a result ⇨ SELECT DISTINCT
Syntax; ⇨ SELECT DISTINCT column_name(s) FROM table_name; --Only unique data

Interview Q&A:
Q: What are some common clauses used with SELECT query in SQL?
A: WHERE clause, ORDER BY clause, GROUP BY clause and HAVING clause

Check Yourself Questions:



Please write a query for selecting the **unique values of "first_name" column** in the **"student_info"** table.

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT DISTINCT first_name FROM student_info;
```

Please write a query for selecting the **unique values of the "last_name" column** in the **"student_info"** table.

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT DISTINCT last_name FROM student_info;
```

Please write a query for selecting the **unique values of "state_name" column** in the **"student_info"** table.

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT DISTINCT state_name FROM student_info;
```

Please write a query for selecting the **unique values of both "state_name" and "county_name" columns** in the **"student_info"** table.

Answer: (penalty regime: 0, 10, 20, ... %)

```sql
SELECT DISTINCT state_name, county_name FROM student_info;
```

Please write a query for selecting the **unique values of both "field" and "start_date" columns** in the **"student_info"** table.

Answer: (penalty regime: 0, 10, 20, ... %)

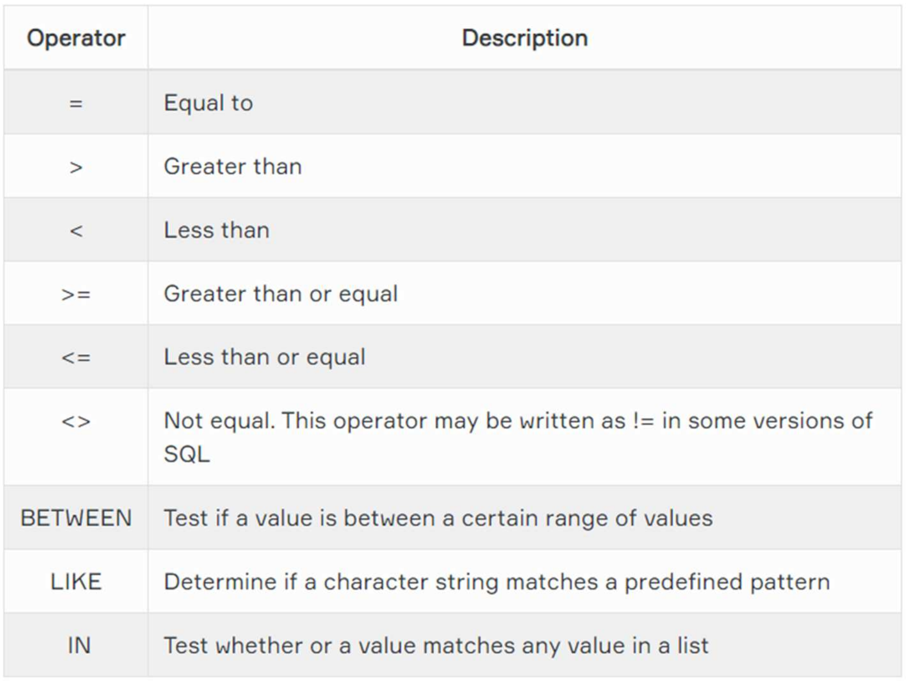
```sql
SELECT DISTINCT field, start_date FROM student_info;
```

4. WHERE & LIMIT Clauses

Introduction

WHERE clause is used to filter records. ⇨ define a specific search condition.
Syntax; ⇨ SELECT column_name(s) FROM table_name WHERE condition(s);
Note: SELECT statement may also be used with some other statements like DELETE and UPDATE.

WHERE Clause – Operators



Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. This operator may be written as != in some versions of SQL
BETWEEN	Test if a value is between a certain range of values
LIKE	Determine if a character string matches a predefined pattern
IN	Test whether or a value matches any value in a list

Example: Assume that; You have a table named student_table as above. If we want to select only the records of which grade is higher than 70 in the result set, then we should write a query like this.
Syntax; ⇨ SELECT * FROM student_table WHERE grade > 70

Example: You want to see only the records of which lesson is Mathematics in the result set.
Syntax; ⇨ SELECT * FROM student_table WHERE lesson = "Mathematics";

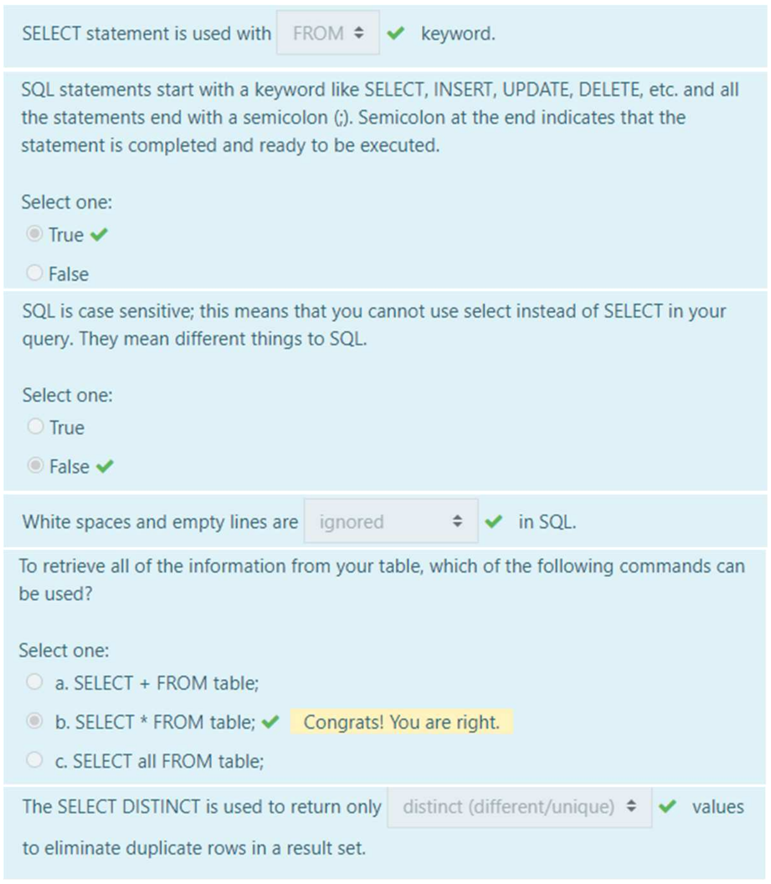
Example: To select only the records of which grade is lower than 70 in the result set.
Syntax; ⇨ SELECT * FROM student_table WHERE grade < 70

LIMIT Clause

LIMIT clause is used to filter records. ⇨ number of rows returned by a query.
Syntax; ⇨ SELECT * FROM student_table LIMIT 3;

Example: Select the students whose grade is higher than 70 and let our query return the first 2 rows.
Syntax; ⇨ SELECT * FROM student_table WHERE grade > 70 LIMIT 2;

Check Yourself Questions:


SELECT statement is used with FROM ✓ keyword.

SQL statements start with a keyword like SELECT, INSERT, UPDATE, DELETE, etc. and all the statements end with a semicolon (;). Semicolon at the end indicates that the statement is completed and ready to be executed.

Select one:
[✓ True] ✓
False

SQL is case sensitive; this means that you cannot use select instead of SELECT in your query. They mean different things to SQL.

Select one:
True
[✓ False] ✓

White spaces and empty lines are ignored ✓ in SQL.

To retrieve all of the information from your table, which of the following commands can be used?

Select one:
a. SELECT + FROM table;
[✓ b. SELECT * FROM table;] ✓ Congrats! You are right.
c. SELECT all FROM table;

The SELECT DISTINCT is used to return only distinct (different/unique) ✓ values to eliminate duplicate rows in a result set.

5. ORDER BY Clause

Order By Clause

ORDER BY keyword sorts the result-set in descending or ascending order.
Syntax; ⇨ SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC;
Syntax; ⇨ SELECT column_name(s) FROM table_name ORDER BY column_name(s) DESC;

Example: I want to sort the *first_name* column from employees table in alphabetical order (A-Z).
Syntax; ⇨ SELECT * FROM employees ORDER BY first_name ASC;

Note: Default sorting order is ASC

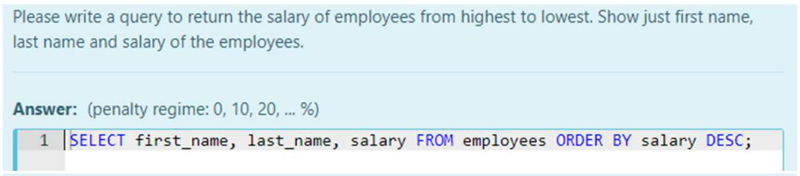
Sorting By Multiple Columns

When need to sort our data by two columns or more.
Syntax; ➪ SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC, column2 ASC|DESC;

ORDER BY Clause with WHERE Clause

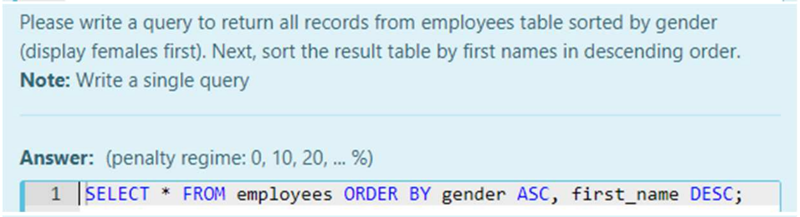
In this part, we will use ORDER BY with the WHERE clause.
Syntax; ➪ SELECT column_name(s) FROM table_name WHERE condition ORDER BY column_name(s);

Check Yourself Questions:


Please write a query to return the salary of employees from highest to lowest. Show just first name, last name and salary of the employees.

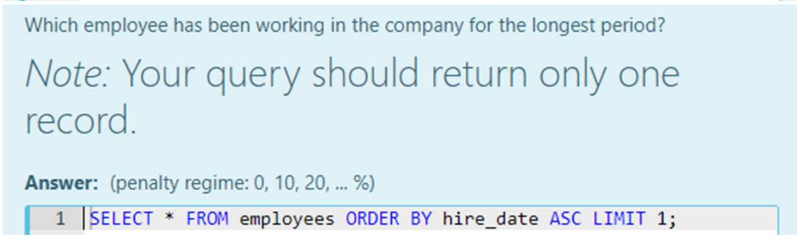
Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT first_name, last_name, salary FROM employees ORDER BY salary DESC;
```


Please write a query to return all records from employees table sorted by gender (display females first). Next, sort the result table by first names in descending order.
Note: Write a single query

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees ORDER BY gender ASC, first_name DESC;
```


Which employee has been working in the company for the longest period?

Note: Your query should return only one record.

Answer: (penalty regime: 0, 10, 20, ... %)

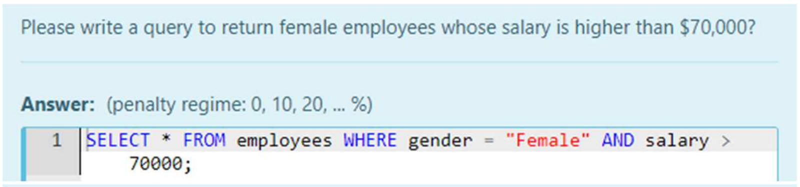
```
SELECT * FROM employees ORDER BY hire_date ASC LIMIT 1;
```

6. AND, OR & NOT Operators

AND, OR & NOT Operators

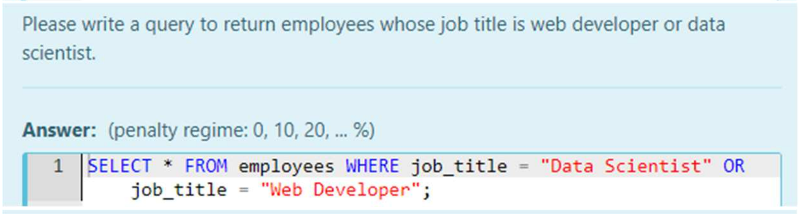
WHERE clause can be combined with AND, OR & NOT operators.
Syntax; ➪ SELECT * FROM employees WHERE job_title = 'Data Scientist' AND gender = 'Male';
Syntax; ➪ SELECT * FROM employees WHERE job_title = 'Data Scientist' OR gender = 'Male';
Syntax; ➪ SELECT * FROM employees WHERE NOT gender = 'Female';

Check Yourself Questions:


Please write a query to return female employees whose salary is higher than $70,000?

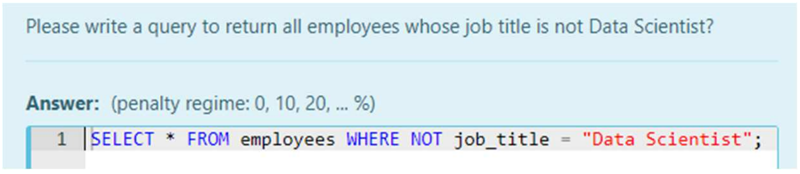
Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees WHERE gender = "Female" AND salary >
    70000;
```


Please write a query to return employees whose job title is web developer or data scientist.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees WHERE job_title = "Data Scientist" OR
    job_title = "Web Developer";
```


Please write a query to return all employees whose job title is not Data Scientist?

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees WHERE NOT job_title = "Data Scientist";
```

7. BETWEEN OPERATOR

Introduction

BETWEEN operator is used for comparison in WHERE clauses.
Syntax; ⇨ WHERE test_expression BETWEEN low_expression AND high_expression
Syntax; ⇨ WHERE test_expression >= low_expression AND test_expression <= high_expression

Example: If we need to find the names of the employees with salary amounts between $80,000 and $90,000, we can use the BETWEEN comparison operator to write:
Syntax; ⇨ SELECT * FROM employees WHERE salary BETWEEN 80000 AND 90000;
Syntax; ⇨ SELECT * FROM employees WHERE salary >= 80000 AND salary <= 90000;

NOT BETWEEN

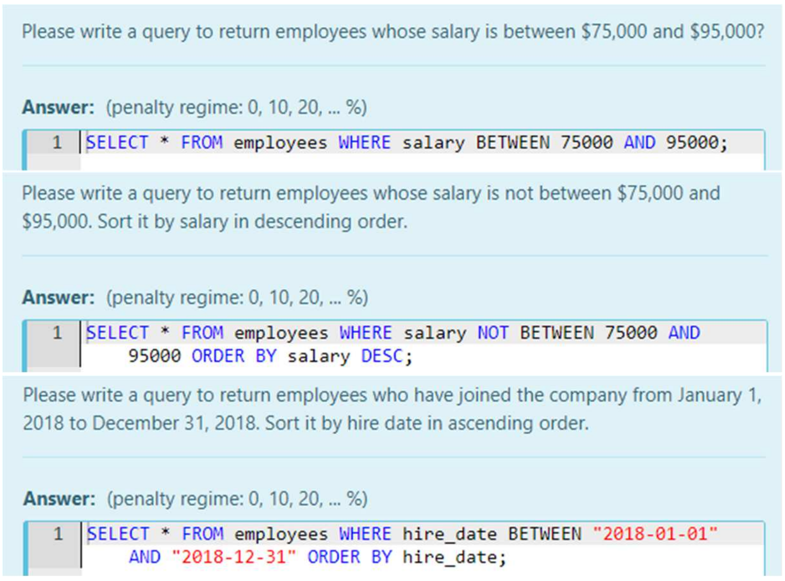
NOT BETWEEN to negate the result of the BETWEEN operator.
Syntax; ⇨ WHERE test_expression NOT BETWEEN low_expression AND high_expression

Example: For instance, you need to find the employees whose salary is not between $80,000 and $90,000. Here is the query:
Syntax; ⇨ SELECT * FROM employees WHERE salary NOT BETWEEN 80000 AND 90000;
Syntax; ⇨ SELECT * FROM employees WHERE salary < 80000 OR salary > 90000;

BETWEEN with Date Example

It's also possible to use the BETWEEN operator with dates.
Important*:* Please enclose your date values with single quote (') and use YYYY-MM-DD date format in your query.
Syntax; ⇨ SELECT * FROM employees WHERE hire_date BETWEEN '2018-06-01' AND '2019-03-31' ORDER BY hire_date;

Check Yourself Questions:



Please write a query to return employees whose salary is between $75,000 and $95,000?

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees WHERE salary BETWEEN 75000 AND 95000;
```

Please write a query to return employees whose salary is not between $75,000 and $95,000. Sort it by salary in descending order.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees WHERE salary NOT BETWEEN 75000 AND
    95000 ORDER BY salary DESC;
```

Please write a query to return employees who have joined the company from January 1, 2018 to December 31, 2018. Sort it by hire date in ascending order.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM employees WHERE hire_date BETWEEN "2018-01-01"
    AND "2018-12-31" ORDER BY hire_date;
```

8. IN Operator

Introduction

IN operator is used to determine whether a value matches any value in a list.
Syntax; ⇨ SELECT * FROM employees WHERE job_title = 'Data Scientist' OR job_title = 'Business Analyst';
Syntax; ⇨ SELECT * FROM employees WHERE job_title IN ('Data Scientist', 'Business Analyst');

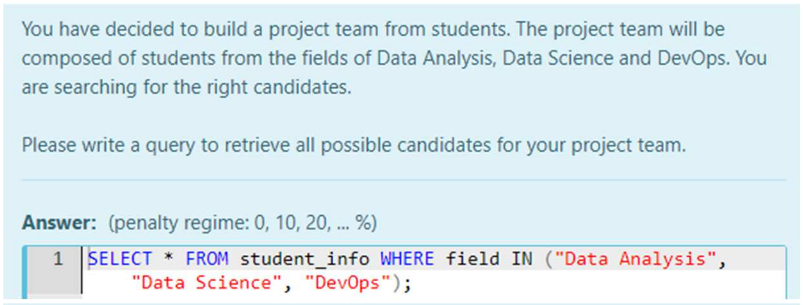
An Extended Value List

Syntax; ⇨ SELECT * FROM employees WHERE job_title IN ('Data Scientist', 'Business Analyst', 'Project Manager', 'Web Developer');

OT IN Operator

Syntax; ⇨ SELECT * FROM employees WHERE job_title NOT IN ('Data Scientist', 'Business Analyst', 'Project Manager', 'Web Developer');

Check Yourself Questions:

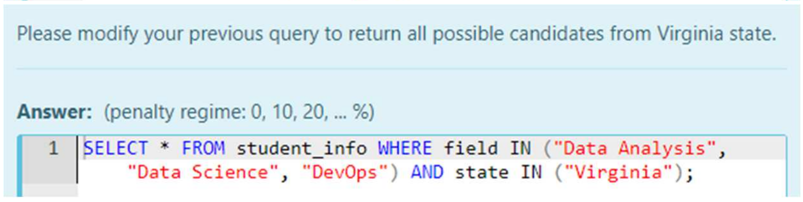


You have decided to build a project team from students. The project team will be composed of students from the fields of Data Analysis, Data Science and DevOps. You are searching for the right candidates.

Please write a query to retrieve all possible candidates for your project team.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM student_info WHERE field IN ("Data Analysis",
    "Data Science", "DevOps");
```



Please modify your previous query to return all possible candidates from Virginia state.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT * FROM student_info WHERE field IN ("Data Analysis",
    "Data Science", "DevOps") AND state IN ("Virginia");
```

9. LIKE Operator

Introduction

LIKE Operator used - We can use when we know just a few letters for desired value/data.
Syntax; ⇨ SELECT column_name(s) FROM table_name WHERE column_1 LIKE pattern;
Percent (%): The % character matches any sequence of zero or more characters.
Underscore (_): The _ character matches any single character.

Examples

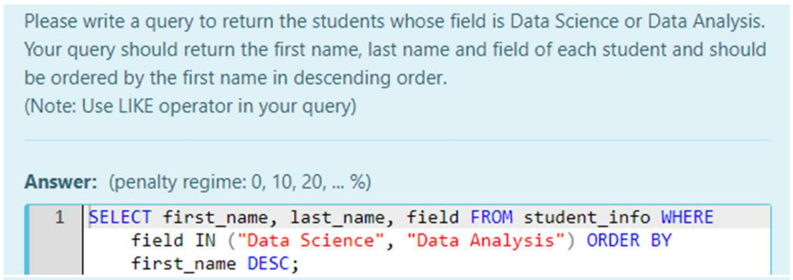
Example: We were trying to recall the county name which starts with "Wo".
Syntax; ⇨ SELECT * FROM student_info WHERE county LIKE 'Wo%';

Example: Suppose that we try to find front-end and back-end developers in our student_info table.
Syntax; ⇨ SELECT * FROM student_info WHERE field LIKE '%Developer';

Note: However, if you want to make LIKE operator case-sensitive, we need to use PRAGMA statement.
Syntax; ⇨ PRAGMA case_sensitive_like = true;
SELECT * FROM student_info WHERE field LIKE '%developer'; --Output of this query is a blank

Example: For instance, the following query finds the employee whose first_name is "Elvis".
Syntax; ⇨ SELECT first_name FROM employees WHERE first_name LIKE 'El_is';

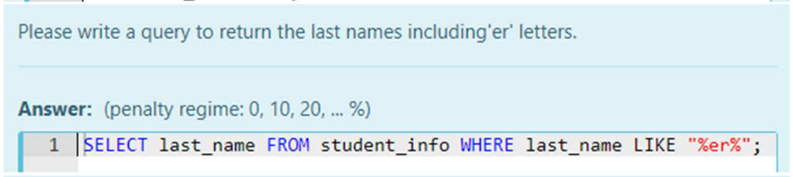
Check Yourself Questions:



Please write a query to return the students whose field is Data Science or Data Analysis. Your query should return the first name, last name and field of each student and should be ordered by the first name in descending order.
(Note: Use LIKE operator in your query)

Answer: (penalty regime: 0, 10, 20, ... %)

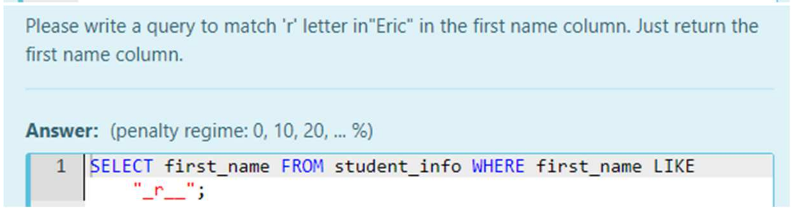
```
SELECT first_name, last_name, field FROM student_info WHERE
    field IN ("Data Science", "Data Analysis") ORDER BY
    first_name DESC;
```



Please write a query to return the last names including'er' letters.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT last_name FROM student_info WHERE last_name LIKE "%er%";
```



Please write a query to match 'r' letter in"Eric" in the first name column. Just return the first name column.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT first_name FROM student_info WHERE first_name LIKE
    "_r__";
```