

# SQL Tutorial



# SQL

# Agenda



Creating, Using, and Dropping a Database in SQL

Data Types in SQL

Constraints in SQL

Create Table

Select

Select Distinct


Where

AND, OR & NOT Operator

Like & Between Operator



# Creating a Database: Syntax

A cartoon illustration of a young boy with brown hair, wearing blue-rimmed glasses, an orange jacket, and a dark scarf. He is holding an open book in his left hand and pointing with his right hand towards a thought bubble. The thought bubble is white with a grey outline and contains the text "Let's see how to **create** a database!".

Let's see how  
to **create** a  
database!

```
CREATE DATABASE databasename;
```


# Using a Database: Syntax



Let's see how  
to **use** a  
database!

```
USE [DatabaseName];
```

# Dropping a Database: Syntax

A cartoon character with brown hair, wearing blue-rimmed glasses, an orange jacket, and a dark scarf. He is holding an open book and pointing with a yellow stick towards a thought bubble.

Let's see how  
to **drop** a  
database!

```
DROP DATABASE databasename;
```

# Data Types in SQL

Data types define what type of data a column can hold.

e_id	e_name	e_salary	e_age	e_gender	e_dept
1	Sam	95000	45	Male	Operations
2	Bob	80000	21	Male	Support
3	Anne	125000	25	Female	Analytics
4	Julia	73000	30	Female	Analytics
5	Matt	159000	33	Male	Sales
6	Jeff	112000	27	Male	Operations

Integer

Character



# Different Data Types in SQL

# Numerical Data Types



Data Type	Range
bigint	$-9223372036854775808 \leftrightarrow 9223372036854775808$
int	$-2147483648 \leftrightarrow 2147483647$
smallint	$-32768 \leftrightarrow -32767$
tinyint	$0 \leftrightarrow 255$
decimal(s,d)	$-10^{38} + 1 \leftrightarrow 10^{38} - 1$



# Character Data Types



Data Type	Range
char(s)	255 Characters
varchar(s)	255 Characters
text	65,535 Characters

# Date and Time Data Types



Data Type	Format
date	YYYY-MM-DD
time	HH:MM:SS
Year	YYYY

# Constraints in SQL



Constraints are used to specify rules for data in a table

Not Null

Default

Unique

Primary Key

# Not Null Constraint



Not Null constraint ensures that a column cannot have a Null value

No null values

e_id	e_name	e_salary	e_age	e_gender	e_dept
1	Sam	95000	45	Male	Operations
2	Bob	80000	21	Male	Support
3	Anne	125000	25	Female	Analytics
4	Julia	73000	30	Female	Analytics
5	Matt	159000	33	Male	Sales
6	Jeff	112000	27	Male	Operations

# Default Constraint



Default constraint sets a default value for a column when no value is specified

	E_id	E_name	E_salary	E_gender	E_dept
1	1	Sam	85000	Male	Analytics
2	2	Anne	85000	Male	Analytics
3	3	Julia	85000	Female	Analytics

Default values

Default values

# Unique Constraint



Unique constraint ensures that all values in a column are different

e_id	e_name	e_salary	e_age	e_gender	e_dept
1	Sam	95000	45	Male	Operations
2	Bob	80000	21	Male	Support
3	Anne	125000	25	Female	Analytics
4	Julia	73000	30	Female	Analytics
5	Matt	159000	33	Male	Sales
6	Jeff	112000	27	Male	Operations

Unique values





# Primary Key Constraint



Primary Key constraint uniquely identifies each record in a table

Not Null + Unique

Primary Key

e_id	e_name	e_salary	e_age	e_gender	e_dept
1	Sam	95000	45	Male	Operations
2	Bob	80000	21	Male	Support
3	Anne	125000	25	Female	Analytics
4	Julia	73000	30	Female	Analytics
5	Matt	159000	33	Male	Sales
6	Jeff	112000	27	Male	Operations

# Foreign Key Constraint



- The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
- The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

O_Id	Order No	P_Id
1	77895	3
2	44678	3
3	22456	2
4	24562	1

```
CREATE TABLE Orders ( O_Id int NOT NULL PRIMARY KEY, OrderNo int NOT NULL, P_Id int FOREIGN KEY REFERENCES Persons(P_Id) )
```

# Create Table



Name the table



Define the  
columns



Assign data  
types to  
columns

# Create Table Statement: Syntax



```
CREATE TABLE table_name(  
column1 datatype,  
column2 datatype,  
column3 datatype,  
.....  
columnN datatype,  
PRIMARY KEY(column_x) );
```

# Select Statement: Syntax



Let's **select**  
some values  
from the **table**!

```
SELECT column1, column2, columnN  
FROM table_name;
```

# Select Distinct: Syntax



Select **Distinct** is used to select only distinct values from our columns

```
SELECT DISTINCT column1,  
column2, columnN FROM table_name;
```



# Where Clause

Where clause is used to extract records which satisfy a condition.



Age>60



Occupation="Doctor"

# Where Clause: Syntax



Let's filter  
records with  
**where** clause!

```
SELECT column1, column2, columnN  
FROM table_name WHERE [condition]
```

# AND Operator

AND operator displays records if all the conditions separated by AND are TRUE.




Age>60

AND



Occupation="Doctor"

# AND Operator: Syntax

A cartoon illustration of a male teacher with brown hair, wearing blue-rimmed glasses, an orange jacket, a dark scarf, and dark pants. He is holding an open book in his left hand and a yellow pointer in his right hand, pointing towards a thought bubble.

Let's impose  
multiple  
conditions  
with '**AND**'!

```
SELECT column1, column2, columnN  
FROM table_name WHERE  
[condition1] AND [condition2]...AND  
[conditionN];
```

# OR Operator

OR operator displays records if any of the conditions separated by OR is TRUE.




Occupation="Software Engineer"

OR



Occupation="Doctor"

# OR Operator: Syntax

A cartoon character of a male teacher with brown hair, wearing blue-rimmed glasses, an orange jacket over a dark scarf, and dark pants. He is holding an open book in his left hand and a yellow pointer in his right hand, pointing towards a thought bubble.

Let's impose  
multiple  
conditions  
with '**OR**'!

```
SELECT column1, column2, columnN  
FROM table_name WHERE  
[condition1] OR [condition2]...OR  
[conditionN];
```



# NOT Operator

NOT operator displays a record if the condition is NOT TRUE.

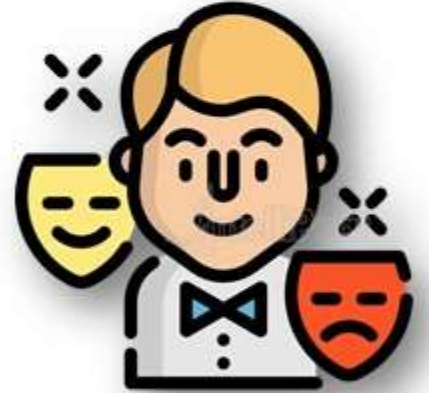
NOT



Occupation="Software Engineer"



Occupation="Doctor"



Occupation="Actor"

# NOT Operator: Syntax



Let's extract  
records where  
the condition  
is **NOT TRUE**

```
SELECT column1, column2, columnN  
FROM table_name WHERE NOT  
[condition];
```

# LIKE Operator

LIKE Operator is used to extract records where a particular pattern is present.

John



Johnathon



Johnny




Marcus

# Wild Card Characters



# LIKE Operator: Syntax

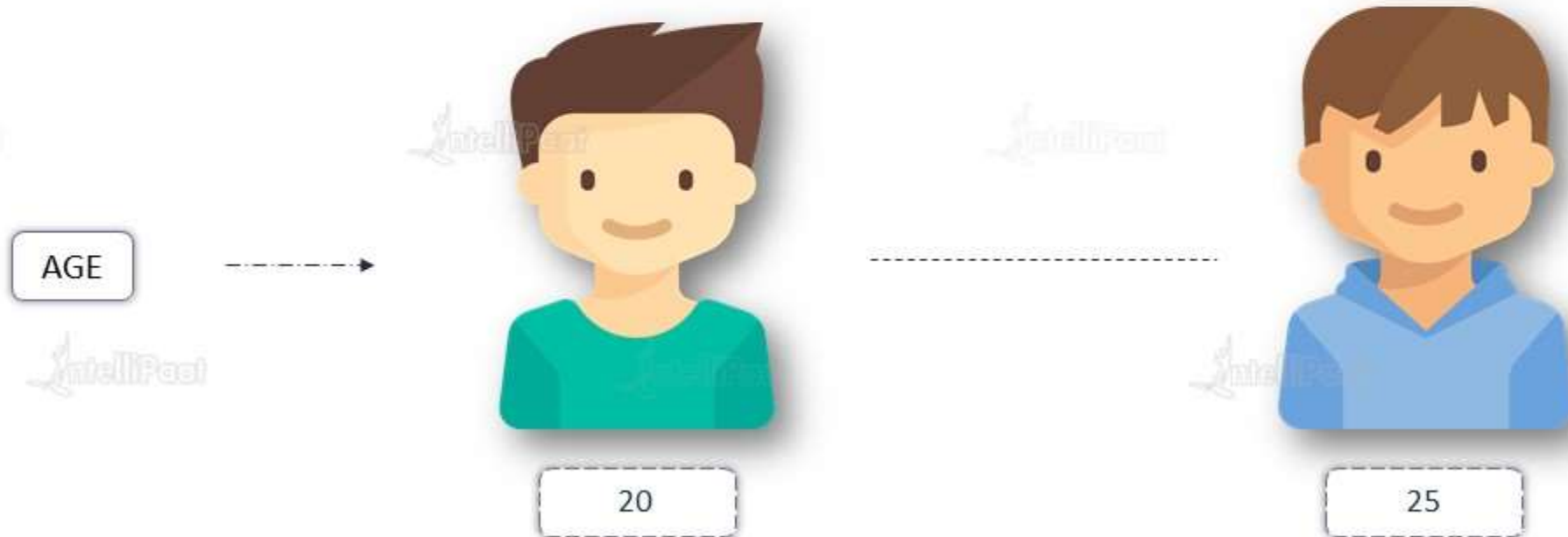
A cartoon character with brown hair, wearing blue-rimmed glasses, an orange jacket, a dark scarf, and dark pants. He is holding an open orange book in his left hand and a yellow pencil in his right hand, pointing it towards a thought bubble. The thought bubble is white with a grey outline and contains the text "Let's extract records where the pattern matches!".

Let's extract  
records where  
the pattern  
matches!

```
SELECT col_list FROM table_name  
WHERE column_N LIKE '_XXXX%';
```


# BETWEEN Operator

BETWEEN Operator is used to select values within a given range.





# BETWEEN Operator: Syntax

A cartoon illustration of a young boy with brown hair, wearing blue-rimmed glasses, an orange jacket, and a dark scarf. He is holding an open book and pointing with a yellow stick towards a thought bubble.

Let's extract  
values  
between a  
given range

```
SELECT col_list FROM table_name  
WHERE column_N BETWEEN val1 AND  
val2;
```

# Quiz

Which of these is the syntax to create a database with the name 'Sparta'

**A**

Create Sparta

**B**

Create Database Sparta

**C**

Build Sparta

**D**

Build Database Sparta



Which of these is the syntax to create a database with the name 'Sparta'

**A**

Create Sparta

**B**

Create Database Sparta

**C**

Build Sparta

**D**

Build Database Sparta



What is the range of 'smallint'?

**A**

0 <-> 255

**B**

-2147483648 <-> 2147483647

**C**

-32768 <-> -32767

**D**

0 <-> 32767



What is the range of 'smallint'?

**A**

0 <-> 255

**B**

-2147483648 <-> 2147483647

**C**

-32768 <-> -32767

**D**

0 <-> 32767





What is the purpose of 'WHERE' clause in Mysql?

**A**

Filtering out unwanted rows from result set

**B**

Filtering out unwanted columns from result set

**C**

Filtering out unwanted rows and columns from result set

**D**

None of the mentioned



What is the purpose of 'WHERE' clause in Mysql?

**A**

Filtering out unwanted rows from result set

**B**

Filtering out unwanted columns from result set

**C**

Filtering out unwanted rows and columns from result set

**D**

None of the mentioned



Which of these is the syntax to select all customers whose name starts with 'Sa%'

**A**

SELECT \* FROM Customers  
WHERE name IN 'Sa%';

**B**

SELECT \* FROM Customers  
WHERE name LIKE 'Sa%';

**C**

SELECT FROM Customers  
WHERE name IS 'Sa%';

**D**

SELECT \* FROM Customers  
WHERE name IS IN 'Sa%';



Which of these is the syntax to select all customers whose name starts with 'Sa%'

**A**

```
SELECT * FROM Customers  
WHERE name IN 'Sa%';
```

**B**

```
SELECT * FROM Customers  
WHERE name LIKE 'Sa%';
```

**C**

```
SELECT FROM Customers  
WHERE name IS 'Sa%';
```

**D**

```
SELECT * FROM Customers  
WHERE name IS IN 'Sa%';
```



**Thank You**