© Copyright 2021 Zenika. All rights reserved

CORE QUERYING LANGUAGES T-SQL

BY

MARCUS CHONG

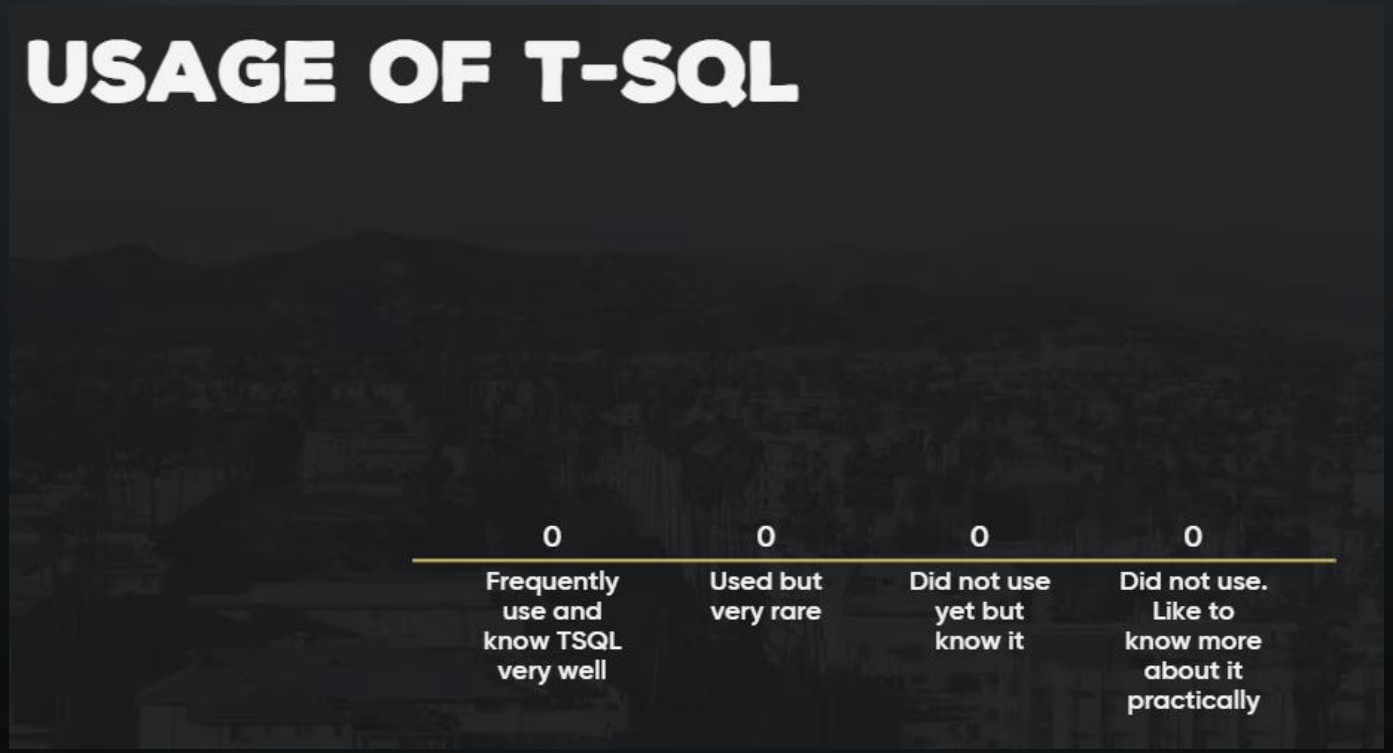
PYTHON DATA ENGINEER @ ZENIKA SINGAPORE

© Copyright 2021 Zenika. All rights reserved

ATTENDANCE

© Copyright 2021 Zenika. All rights reserved

LET ME GET TO KNOW ABOUT YOU



© Copyright 2021 Zenika. All rights reserved

AGENDA

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

ZENIKA CHICKEN RICE SHOP

© Copyright 2021 Zenika. All rights reserved

AS BUSINESS EXPANDS…

- Location
- Inventories
- Purchases
- Sales
- Employees
- Customers
- Membership

© Copyright 2021 Zenika. All rights reserved

PROBLEMS…..

- How do I know which location is selling better?
- Can I assign membership to customer based on certain criteria?
- How do I know the transactions of a customer every month?
- How do I monitor my inventories?
- How do I update customer information?
- How do I delete unnecessary information?

© Copyright 2021 Zenika. All rights reserved

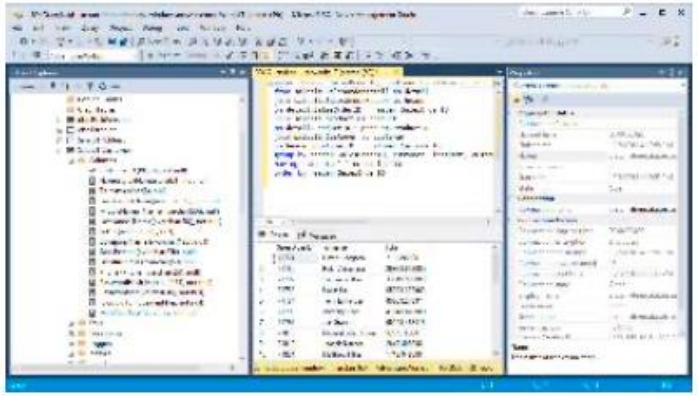
SOLUTION…

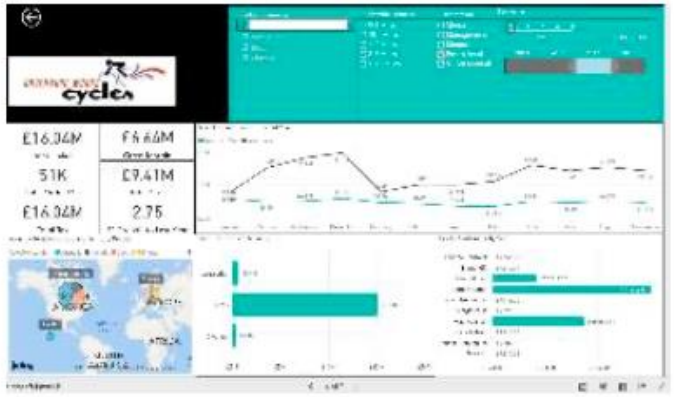
Database

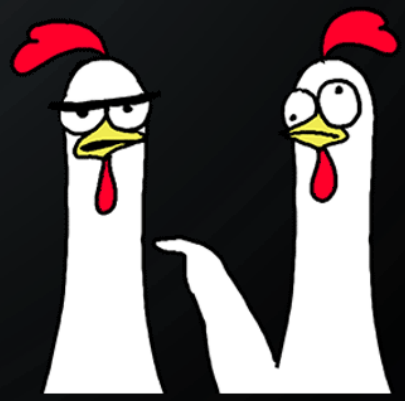
Microsoft SQL Server

Visualisation

Power BI







© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

DATABASE

- A collection of data stored in a format that can easily be accessed

DATABASE MANAGEMENT SYSTEM (DBMS)

- Software application to manage our database

© Copyright 2021 Zenika. All rights reserved

DBMS

Do not understand SQL

Relational

- SQL Server
- MySQL
- Oracle
- IBM DB2
- etc

NoSQL

- MongoDB
- CouchDB
- CouchBase
- Redis
- etc

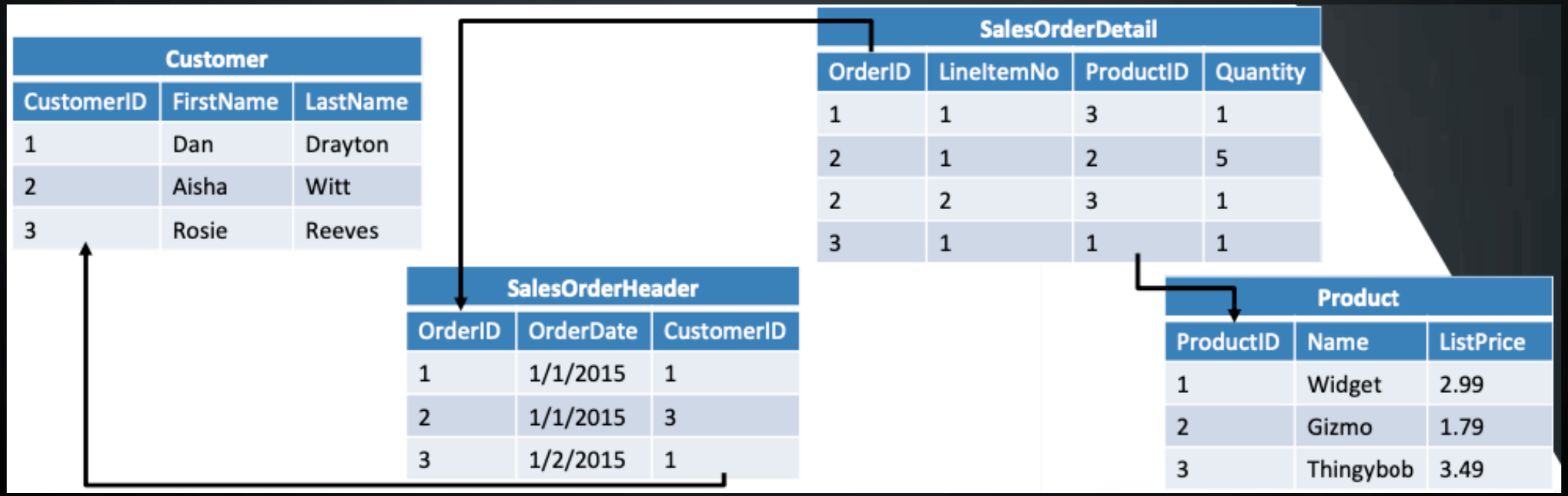
© Copyright 2021 Zenika. All rights reserved

RELATIONAL DATABASE

- Entities are represented as relations (tables), in which their attributes are represented as domains (columns)
- Most relational databases are normalized, with relationships defined between tables through primary and foreign keys

© Copyright 2021 Zenika. All rights reserved

RELATIONAL DATABASE



Customer

CustomerID	FirstName	LastName
1	Dan	Drayton
2	Aisha	Witt
3	Rosie	Reeves

SalesOrderHeader

OrderID	OrderDate	CustomerID
1	1/1/2015	1
2	1/1/2015	3
3	1/2/2015	1

SalesOrderDetail

OrderID	LineItemNo	ProductID	Quantity
1	1	3	1
2	1	2	5
2	2	3	1
3	1	1	1

Product

ProductID	Name	ListPrice
1	Widget	2.99
2	Gizmo	1.79
3	Thingybob	3.49

© Copyright 2021 Zenika. All rights reserved

SCHEMAS AND OBJECT NAMES

- A Schema in SQL is a collection of database objects associated with a database.
- Schemas are separate namespaces/containers for database objects such as tables, views, stored procedures, indexes, etc.
- The schema is owned by one of the following database-level principals: database user, database role, or application role.
- Schema always belong to a single database whereas a database can have single or multiple schemas.
- Fully-qualified names: [server_name.][database_name.][schema_name.]object_name
- Within database context, best practice is to include schema name: schema_name.object_name

```
1  /****** Script for SelectTopNRows command from SSMS  ******/
2  SELECT TOP (1000) CustomerNumber
3        ,CustomerFirstName
4        ,CustomerMiddleName
5        ,CustomerLastName
6        ,CustomerIC
7        ,DateOfBirth
8        ,Location
9    FROM Zenika-Chicken.dbo.tblCustomer
```


Source: https://stackoverflow.com/questions/2221929/what-is-the-importance-of-schema-in-sql-server

© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

STRUCTURED QUERY LANGUAGE (SQL)

- Structured English Query Language (SEQUEL)
- Developed by IBM in 1970s (initially called SEQUEL but changed to SQL later)
- Widely used in industry
- … but many vendors specific implementations (IBM, Oracle, Sybase, Microsoft, Open sources …)
- SQL is a declarative language
  - you tell what you want and the SQL engine decides how

© Copyright 2021 Zenika. All rights reserved

TRANSACT-SQL (T-SQL)

- What's T-SQL?
- Microsoft's implementation
- Query language for SQL Server and Azure SQL Database
- T-SQL expands on the SQL standard to include
  - Procedural programming
  - Local variables
  - String, date and mathematics functions
  - Specific DELETE and UPDATE statements

© Copyright 2021 Zenika. All rights reserved

NEXT

© Copyright 2021 Zenika. All rights reserved

SO YOU ARE HIRED!

OK sure!

Hey Marcus, welcome onboard. Please create a database for location, customers, and customers' transactions.

© Copyright 2021 Zenika. All rights reserved

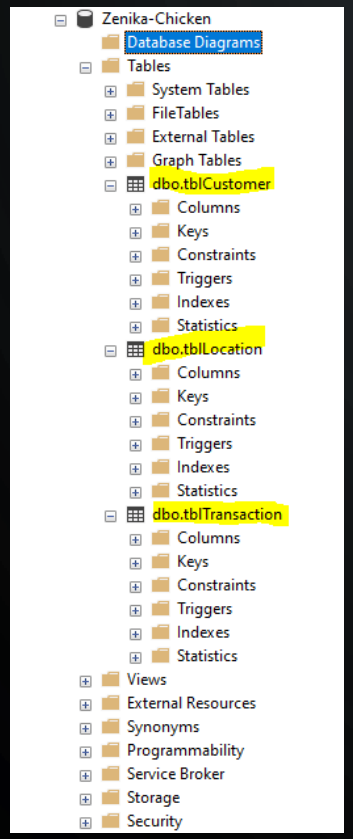
CREATE DATABASE AND TABLES

Continue….

```
1
2   -- Creating tblCustomer
3   create table [dbo].[tblCustomer] (
4       [CustomerNumber] int NOT NULL,
5       [CustomerFirstName] varchar(50) NOT NULL,
6       [CustomerMiddleName] varchar(50) NULL,
7       [CustomerLastName] varchar(50) NOT NULL,
8       [CustomerIC] char(10) NOT NULL,
9       [DateOfBirth] date NOT NULL,
10      [Location] varchar(19) NULL
11  )
12
13  -- Creating tblTransaction
14  create table [dbo].[tblTransaction] (
15      [Amount] smallmoney NOT NULL,
16      [DateOfTransaction] smalldatetime NOT NULL,
17      [CustomerNumber] int NOT NULL
18  )
19
20  -- Creating tblLocation
21  create table [dbo].[tblLocation] (
22      [Location] varchar(19) NULL,
23      [LocationHead] varchar(19) NULL
24  )
```

© Copyright 2021 Zenika. All rights reserved

CREATE DATABASE AND TABLES



tblLocation

Column Name	Data Type	Allow Nulls
Location	varchar(19)	☑
LocationHead	varchar(19)	☑
		☐

tblCustomer

Column Name	Data Type	Allow Nulls
CustomerNumber	int	☐
CustomerFirstName	varchar(50)	☐
CustomerMiddleName	varchar(50)	☑
CustomerLastName	varchar(50)	☐
CustomerIC	char(10)	☐
DateOfBirth	date	☐
Location	varchar(19)	☑
		☐

tblTransaction

Column Name	Data Type	Allow Nulls
Amount	smallmoney	☐
DateOfTransaction	smalldatetime	☐
CustomerNumber	int	☐
		☐

- 5 rows
- Location: North, East, South, West, Central

- 1005 rows
- CustomerNumber 1, 2, 123 - 1125

- 2500 rows
- CustomerNumber 1 - 1125

© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	**05**	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

INSERT STATEMENT

No problem!

Well done Marcus. Now can you please INSERT the data into the database?

© Copyright 2021 Zenika. All rights reserved

INSERT STATEMENT

```
1  INSERT INTO tblCustomer(
2      [CustomerFirstName],
3      [CustomerMiddleName],
4      [CustomerLastName],
5      [CustomerIC],
6      [DateOfBirth],
7      [Location],
8      [CustomerNumber]
9  )
10 VALUES ('Jossef', 'H', 'Wright', 'TX593671R', '19711224', 'Litigation',131)
11
12 INSERT INTO tblCustomer
13 VALUES (1, 'Dylan', 'A', 'Word', 'HN513777D', '19920914', 'Customer Relations'),
14 (2,'Jossef', 'H', 'Wright', 'TX593671R', '19711224', 'Litigation')
```

How about inserting data from my excel file?

© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

SELECT STATEMENTS

That's easy!

Well, can you show me how to read the data?

© Copyright 2021 Zenika. All rights reserved

BASIC SELECT

```sql
-- All columns
SELECT *
FROM tblCustomer;


-- Specific columns
SELECT
    CustomerFirstName,
    CustomerMiddleName,
    CustomerLastName,
    Location
FROM tblCustomer;


-- Aliases
SELECT
    CustomerNumber,
    Amount as 'Customer Spending',
    DateOfTransaction
FROM tblTransaction;
```

© Copyright 2021 Zenika. All rights reserved

SELECT ONLY PART OF THE TABLE

WHERE

Predicates and Operators	Description
= < >	Compares values for equality / non-equality.
IN	Determines whether a specified value matches any value in a subquery or a list.
BETWEEN	Specifies an inclusive range to test.
LIKE	Determines whether a specific character string matches a specified pattern, which can include wildcards.
AND	Combines two Boolean expressions and returns TRUE only when both are TRUE.
OR	Combines two Boolean expressions and returns TRUE if either is TRUE.
NOT	Reverses the result of a search condition.

© Copyright 2021 Zenika. All rights reserved

SELECT ONLY PART OF THE TABLE

Continue….

STRING

```
1  SELECT * FROM tblCustomer
2  WHERE CustomerFirstName = 'Dylan'; -- equal
3
4  SELECT * FROM tblCustomer
5  WHERE CustomerFirstName <> 'Dylan'; -- not equal
6
7  SELECT * FROM tblCustomer
8  WHERE CustomerFirstName > 'Word'; -- alphabet order higher than Word
9
10 SELECT * FROM tblCustomer
11 WHERE CustomerFirstName LIKE '%we%'; -- contains 'we' in first name
12
13 SELECT * FROM tblCustomer
14 WHERE CustomerFirstName LIKE '[r-t]%'; -- start with r to t
15
16 SELECT * FROM tblCustomer
17 WHERE CustomerFirstName LIKE '[^rwt]%'; -- start not with r, w or t
```

© Copyright 2021 Zenika. All rights reserved

SELECT ONLY PART OF THE TABLE

Continue….

NUMBER

```
SELECT TOP (10) CustomerNumber
      ,CustomerFirstName
      ,CustomerMiddleName
      ,CustomerLastName
FROM tblCustomer;

SELECT * FROM tblCustomer
WHERE not CustomerNumber>200;

SELECT * FROM tblCustomer
WHERE CustomerNumber!=200;

SELECT * FROM tblCustomer
WHERE CustomerNumber>=200 AND CustomerNumber<=209;

SELECT * FROM tblCustomer
WHERE NOT (CustomerNumber>=200 AND CustomerNumber<=209);

SELECT * FROM tblCustomer
WHERE CustomerNumber<200 OR CustomerNumber>209;

SELECT * FROM tblCustomer
WHERE CustomerNumber BETWEEN 200 AND 209;

SELECT * FROM tblCustomer
WHERE CustomerNumber NOT BETWEEN 200 AND 209;

SELECT * FROM tblCustomer
WHERE CustomerNumber IN (200, 204, 208);
```

© Copyright 2021 Zenika. All rights reserved

SELECT ONLY PART OF THE TABLE

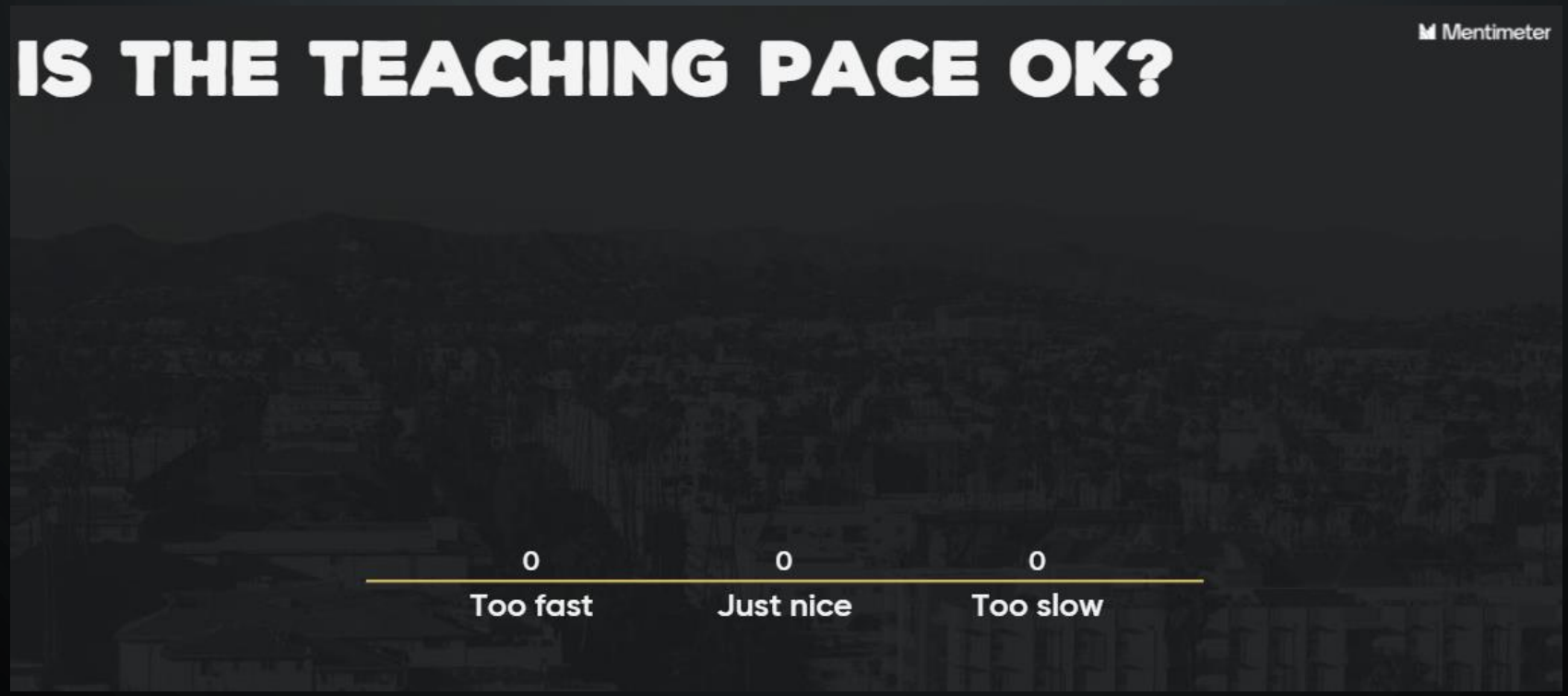
Continue….

DATETIME

```
DECLARE @mydate AS datetime = '2021-05-25';
SELECT
    YEAR(@mydate) AS myYear, -- 2021
    MONTH(@mydate) AS myMonth, -- 5
    DAY(@mydate) AS myDay; -- 25

SELECT DATEFROMPARTS(2021,05,24) AS ThisDate;
-- 2021-05-24

SELECT GETDATE() AS RightNow;

SELECT DATEADD(YEAR,1,'2021-01-01 03:04:05') AS myYear;
-- 2022-01-01 03:04:05.000

SELECT DATEPART(HOUR,'2021-01-01 03:04:05') AS myHour;
-- 3

SELECT DATENAME(weekday, GETDATE()) AS myDay;


SELECT * FROM tblCustomer
WHERE DateOfBirth between '19850101' and '19861231'
-- can also put 1985-01-01 (inclusive 19850101 and 19861231)

SELECT * from tblCustomer
WHERE DateOfBirth >= '19760101' and DateOfBirth < '19870101'

SELECT * FROM tblCustomer
WHERE year(DateOfBirth) = 1967
```

© Copyright 2021 Zenika. All rights reserved

FEEDBACK TIME…



© Copyright 2021 Zenika. All rights reserved

QUESTION 1

© Copyright 2021 Zenika. All rights reserved

QUESTION 2

© Copyright 2021 Zenika. All rights reserved

QUESTION 3

© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

AGGREGATE FUNCTIONS

Sure, I can do that!

That's a good demonstration. Can you tell me the total amount and the total number of transaction of specific customers?

Can you also group the customers based on their Initials?

© Copyright 2021 Zenika. All rights reserved

AGGREGATE FUNCTIONS

Continue….

- Functions that operate on sets, or rows of data
- Summarize input rows
- Example:
  - MAX()
  - MIN()
  - AVG()
  - SUM()
  - COUNT()

```
SELECT
    MAX(AMOUNT) AS HIGHEST,
    MIN(AMOUNT) AS LOWEST,
    AVG(AMOUNT) AS AVERAGE,
    SUM(AMOUNT) AS TOTAL,
    COUNT(AMOUNT) AS NUMBER_OF_TRANSACTION,
    COUNT(*) AS TOTAL_RECORDS
FROM tblTransaction
```

© Copyright 2021 Zenika. All rights reserved

GROUP BY

- GROUP BY creates groups for output rows, according to a unique combination of values specified in the GROUP BY clause
- GROUP BY calculates a summary value for aggregate functions in subsequent phases

```
-- What if I want the total amount and the number
-- of transaction of customer 100 to 199?

-- FROM table transaction
--      WHERE customer number between 100 and 199
--      GROUP BY customer number
--          SELECT customer number
--          SUM amount of transaction
--          COUNT number of transaction
```

```
SELECT
    CustomerNumber,
    COUNT(AMOUNT) as NumberOfTransactions,
    SUM(AMOUNT) as TotalAmount
FROM tblTransaction
WHERE CustomerNumber BETWEEN 100 and 199
GROUP BY CustomerNumber
```

```
-- What if I want the total number of customers with date of birth > 19860101,
-- segregated by Initial of their name?

-- FROM table customer
--      WHERE customer data of birth > 19860101
--      GROUP BY customer's Initial
--          SELECT Initial
--          COUNT number of rows
```

```
SELECT
    LEFT(CustomerLastName,1) AS Initial,
    COUNT(*) AS CountOfInitial
FROM tblCustomer
WHERE DateOfBirth > '19860101'
GROUP BY LEFT(CustomerLastName,1) -- we are not allowed to use alias Initial here
```

© Copyright 2021 Zenika. All rights reserved

HAVING

- HAVING clause provides a search condition that each group must satisfy
- WHERE clause is processed before GROUP BY, HAVING clause is processed after GROUP BY

```
-- What if I want the total amount and the number
-- of transaction of customer 100 to 199?
-- But also with amount of transaction >= 100?

-- FROM table transaction
--      WHERE customer number between 100 and 199
--      GROUP BY customer number
--          HAVING amount of transaction >= 100
--              SELECT customer number
--              SUM amount of transaction
--              COUNT number of transaction
```

```
-- What if I want the total number of customers with date of birth > 19860101,
-- segregated by Initial of their name?
-- But also with count of initial > 1

-- FROM table customer
--      WHERE customer data of birth > 19860101
--      GROUP BY customer's Initial
--          HAVING count of initial > 1
--              SELECT Initial
--              COUNT number of rows
```

```
SELECT
    CustomerNumber,
    COUNT(AMOUNT) as NumberOfTransactions,
    SUM(AMOUNT) as TotalAmount
FROM tblTransaction
WHERE CustomerNumber BETWEEN 100 and 199
GROUP BY CustomerNumber
HAVING SUM(AMOUNT) >=100; -- cannot use TotalAmount
```

```
SELECT
    LEFT(CustomerLastName,1) AS Initial,
    COUNT(*) AS CountOfInitial
FROM tblCustomer
WHERE DateOfBirth > '19860101'
GROUP BY LEFT(CustomerLastName,1)
HAVING COUNT(*) > 1;
```

© Copyright 2021 Zenika. All rights reserved

SORTING

Opps, I forgot about that. Let me fix it.

Not there yet, Marcus. I want the data to be sorted according to the count and the amount!

© Copyright 2021 Zenika. All rights reserved

ORDER BY

```
-- What if I want the total amount and the number
-- of transaction of customer 100 to 199?
-- But also with amount of transaction >= 100?

-- FROM table transaction
--      WHERE customer number between 100 and 199
--      GROUP BY customer number
--          HAVING amount of transaction >= 100
--              SELECT customer number
--              SUM amount of transaction
--              COUNT number of transaction
--                  ORDER BY NumberOfTransactions in descending order and Total Amount
```

```
SELECT
    CustomerNumber,
    COUNT(AMOUNT) as NumberOfTransactions,
    SUM(AMOUNT) as TotalAmount
FROM tblTransaction
WHERE CustomerNumber BETWEEN 100 and 199
GROUP BY CustomerNumber
HAVING SUM(AMOUNT) >=100 -- cannot use TotalAmount
ORDER BY NumberOfTransactions DESC, TotalAmount;
```

```
-- What if I want the total number of customers with date of birth > 19860101,
-- segregated by Initial of their name?
-- But also with count of initial > 1

-- FROM table customer
--      WHERE customer data of birth > 19860101
--      GROUP BY customer's Initial
--          HAVING count of initial > 1
--              SELECT Initial
--              COUNT number of rows
--                  ORDER BY CountOfInitial in descending order
```

```
SELECT
    LEFT(CustomerLastName,1) AS Initial,
    COUNT(*) AS CountOfInitial
FROM tblCustomer
WHERE DateOfBirth > '19860101'
GROUP BY LEFT(CustomerLastName,1)
HAVING COUNT(*) > 1
ORDER BY CountOfInitial DESC;
```

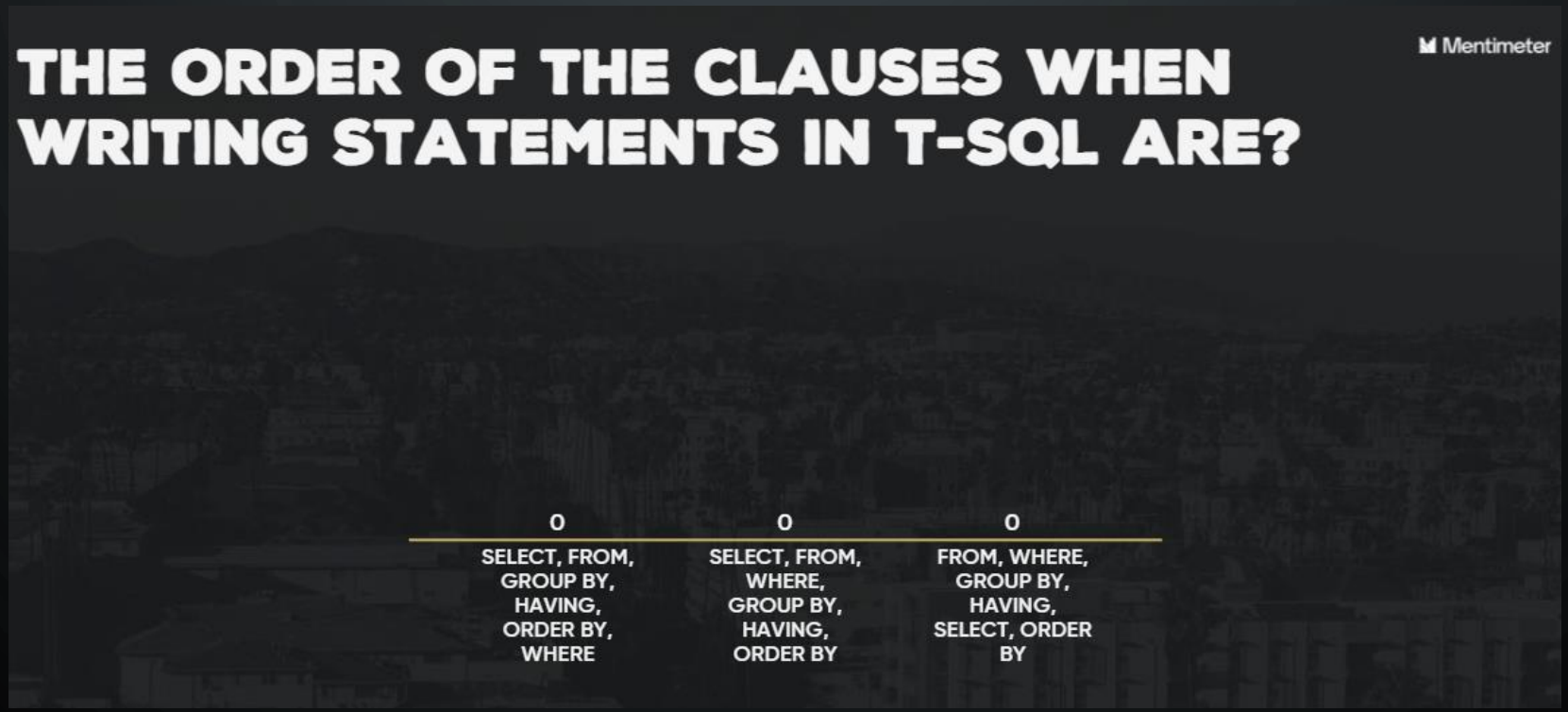
© Copyright 2021 Zenika. All rights reserved

REMEMBER THIS! THIS IS HOW SQL EXECUTES TASK

	Element	Expression	Role
5	SELECT	<select list>	Defines which columns to return
1	FROM	<table source>	Defines table(s) to query
2	WHERE	<search condition>	Filters rows using a predicate
3	GROUP BY	<group by list>	Arranges rows by groups
4	HAVING	<search condition>	Filters groups using a predicate
6	ORDER BY	<order by list>	Sorts the output

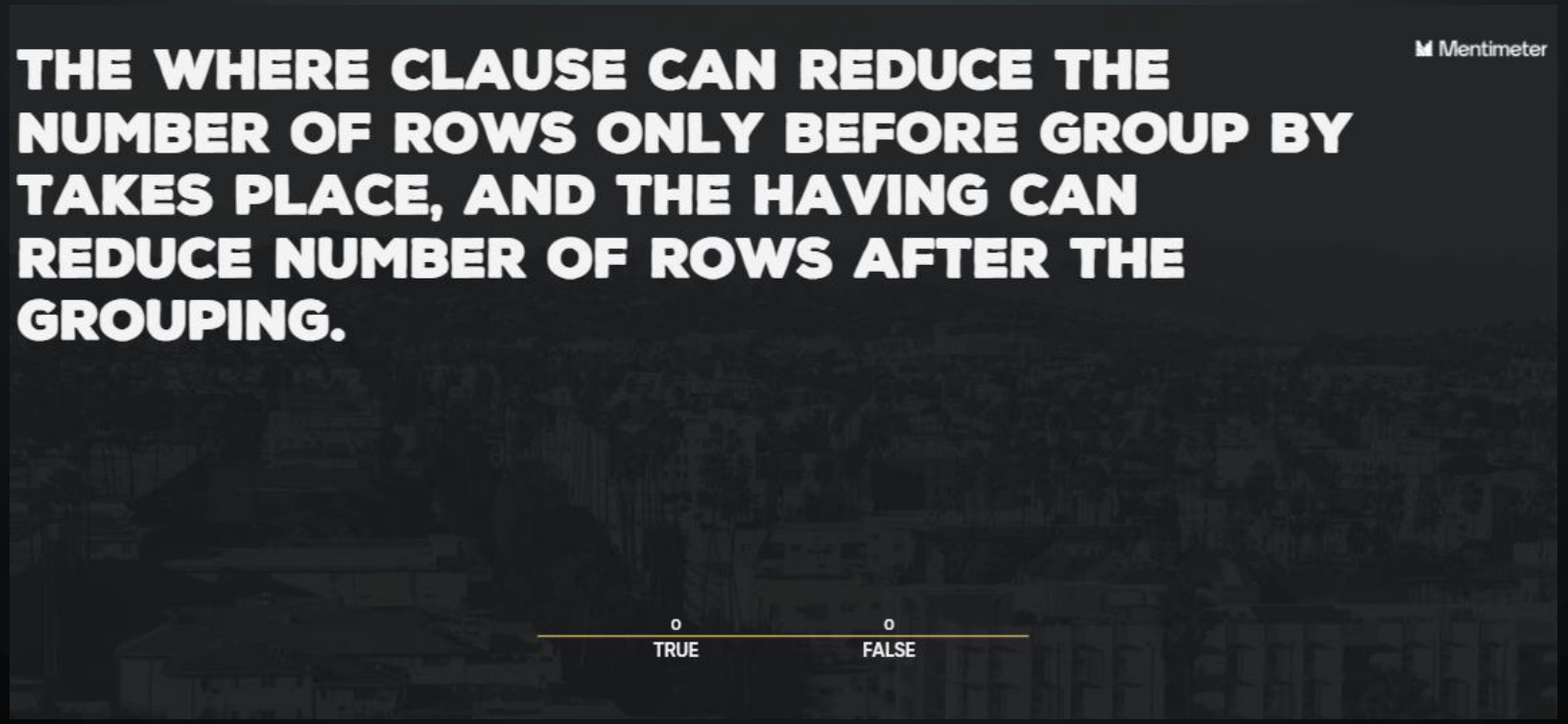
© Copyright 2021 Zenika. All rights reserved

QUESTION 4



© Copyright 2021 Zenika. All rights reserved

QUESTION 5



© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	**08**	**Join Tables**
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

JOIN TABLES

Ok boss, you should have told me earlier…

Hey Marcus, you've created 3 tables. Can you please show the columns of these tables together?

© Copyright 2021 Zenika. All rights reserved

JOIN TABLE

Continue….

- Combine rows from multiple tables by specifying matching criteria
- Usually based on primary key – foreign key relationships

tblLocation

Column Name	Data Type	Allow Nulls
Location	varchar(19)	☑
LocationHead	varchar(19)	☑
		☐

tblCustomer

Column Name	Data Type	Allow Nulls
CustomerNumber	int	☐
CustomerFirstName	varchar(50)	☐
CustomerMiddleName	varchar(50)	☑
CustomerLastName	varchar(50)	☐
CustomerIC	char(10)	☐
DateOfBirth	date	☐
Location	varchar(19)	☑
		☐

tblTransaction

Column Name	Data Type	Allow Nulls
Amount	smallmoney	☐
DateOfTransaction	smalldatetime	☐
CustomerNumber	int	☐
		☐

© Copyright 2021 Zenika. All rights reserved

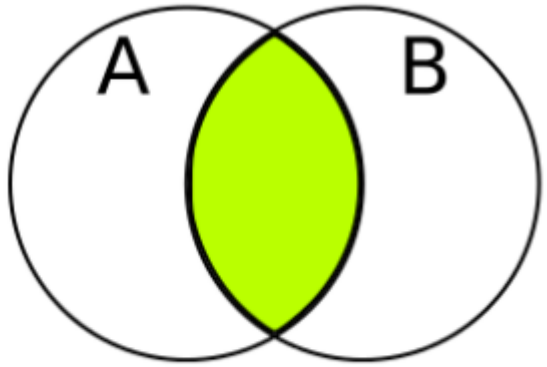
JOIN TABLE

Continue….

- Inner join
- Left join (inclusive)
- Left join (exclusive)
- Right join (inclusive)
- Right join (exclusive)
- Full join (inclusive)
- Full join (exclusive)
- Joining more than 1 tables
- Cross join

© Copyright 2021 Zenika. All rights reserved

INNER JOIN



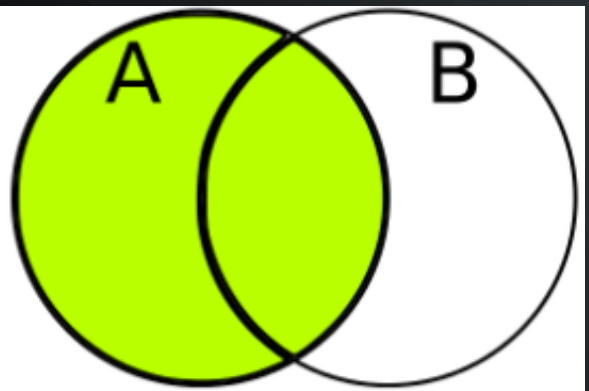

```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

```sql
-- Inner join -- 898 rows
-- Only customers that exist in both tables
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

LEFT JOIN (INCLUSIVE)



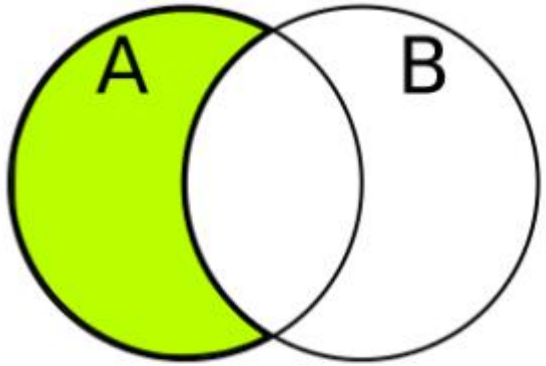

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
```

```sql
-- Left join (inclusive) -- 1005 rows
-- Customers exist in both tables + customers in table A not in B
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
LEFT JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

LEFT JOIN (EXCLUSIVE)



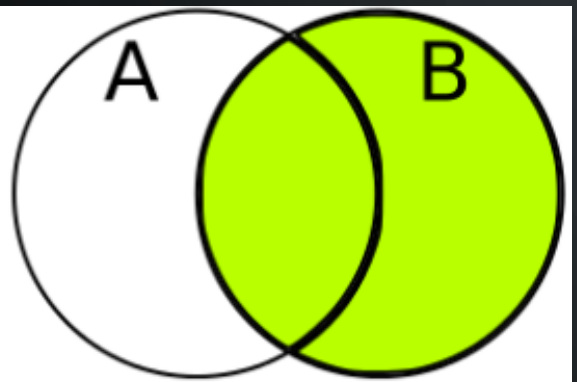

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

```
-- Left join (exclusive) - 107 rows
-- Customer in Table A, but not in table B
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
LEFT JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
WHERE B.CustomerNumber IS NULL
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

RIGHT JOIN (INCLUSIVE)



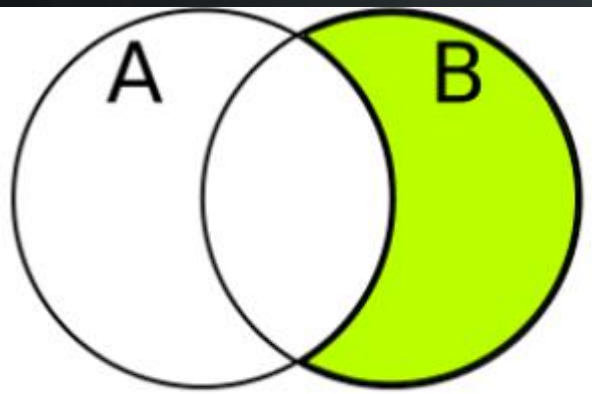

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
```

```
-- Right join (inclusive) -- 1002 rows
-- Customers exist in both tables + customer in table B not in A (3 to 122)
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
RIGHT JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

RIGHT JOIN (EXCLUSIVE)



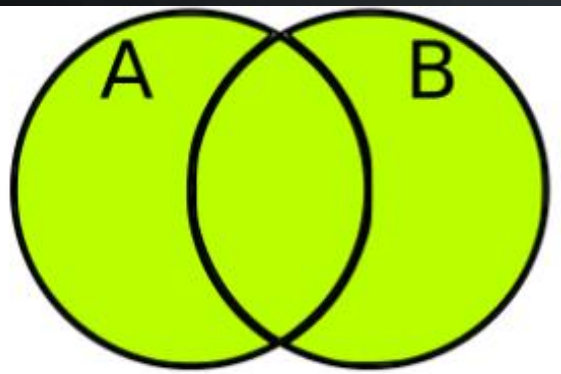

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE A .key IS NULL
```

```sql
-- Right join (exclusive) - 104 rows
-- Customer in Table B, but not in A (3 to 122)
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
RIGHT JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
WHERE A.CustomerNumber IS NULL
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

FULL JOIN (INCLUSIVE)



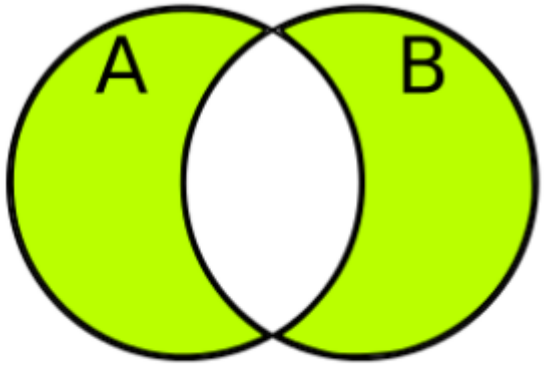

```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
```

```sql
-- Full join (inclusive) -- 1109 rows
-- Equivalent to Left join inclusive + Right join exclusive
-- Equivalent to Right join inclusive + Left join exclusive
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
FULL JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

FULL JOIN (EXCLUSIVE)




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
WHERE B.key IS NULL
OR A.key IS NULL
```

```
-- Full join (exclusive)
-- Equivalent to Left Join Exclusive + Right Join Exclusive
SELECT
    A.CustomerNumber,
    B.CustomerNumber,
    CustomerFirstName,
    CustomerLastName,
    SUM(Amount) AS TotalTransactionAmount
FROM tblCustomer AS A
FULL JOIN tblTransaction AS B
    ON A.CustomerNumber = B.CustomerNumber
WHERE A.CustomerNumber IS NULL or B.CustomerNumber IS NULL
GROUP BY A.CustomerNumber, B.CustomerNumber, CustomerFirstName, CustomerLastName
ORDER BY A.CustomerNumber;
```


Image source: https://medium.com/@gianpaul.r/sql-joins-2caf5487316d

© Copyright 2021 Zenika. All rights reserved

JOIN MORE THAN 1 TABLE

tblLocation

Column Name	Data Type	Allow Nulls
Location	varchar(19)	☑
LocationHead	varchar(19)	☑
		☐

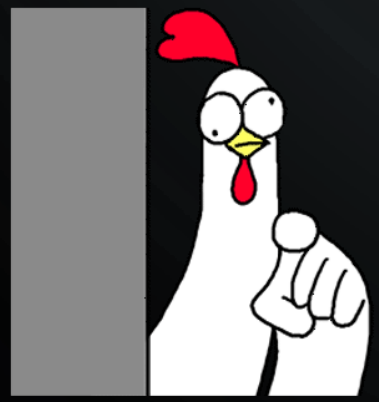
tblCustomer

Column Name	Data Type	Allow Nulls
CustomerNumber	int	☐
CustomerFirstName	varchar(50)	☐
CustomerMiddleName	varchar(50)	☑
CustomerLastName	varchar(50)	☐
CustomerIC	char(10)	☐
DateOfBirth	date	☐
Location	varchar(19)	☑
		☐

tblTransaction

Column Name	Data Type	Allow Nulls
Amount	smallmoney	☐
DateOfTransaction	smalldatetime	☐
CustomerNumber	int	☐
		☐

Please tell me the total sales of each Location (Central, North, East, West, South).



```sql
-- Join more than 1 tables
-- total sales of each Location
SELECT
    L.Location as Area,
    L.LocationHead as Manager,
    SUM(Amount) as TotalTransactionAmount
FROM tblLocation AS L
LEFT JOIN tblCustomer AS C
    ON L.Location = C.Location
LEFT JOIN tblTransaction AS T
    ON C.CustomerNumber = T.CustomerNumber
GROUP BY L.Location, L.LocationHead;
```

© Copyright 2021 Zenika. All rights reserved

CROSS JOIN

Create different combinations between two tables

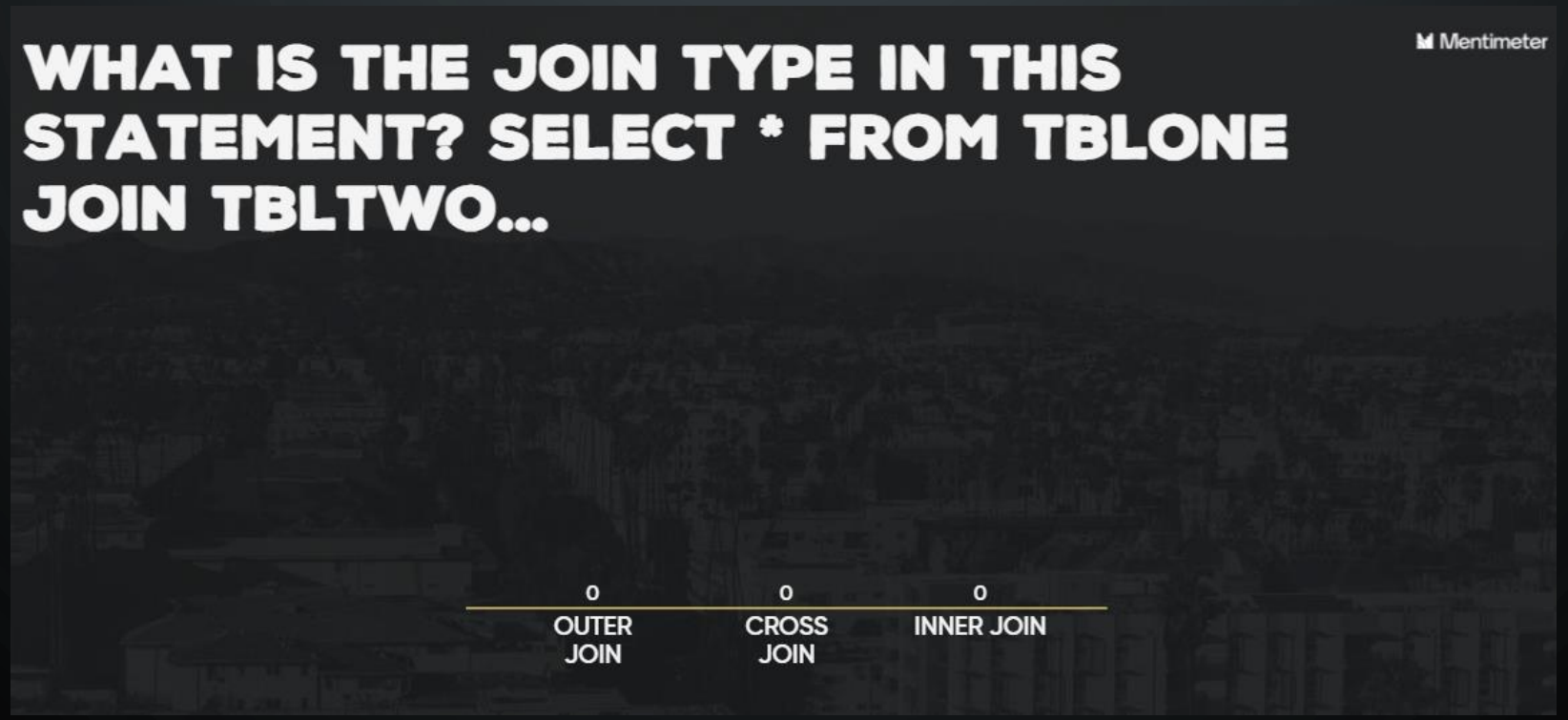
```
-- CROSS JOIN
CREATE TABLE tblWorkday (
    workday varchar(10) NOT NULL
);

INSERT INTO tblWorkday VALUES
('Monday'), ('Tuesday'), ('Wednesday'), ('Thursday'), ('Friday');

SELECT * FROM tblWorkday;

SELECT
    T1.LocationHead,
    T2.workday
FROM tblLocation as T1
CROSS JOIN tblWorkday as T2
ORDER BY T1.LocationHead;
```

Results | Messages

	LocationHead	workday
1	Andrew	Monday
2	Andrew	Tuesday
3	Andrew	Wednesday
4	Andrew	Thursday
5	Andrew	Friday
6	Bryan	Monday
7	Bryan	Tuesday
8	Bryan	Wednesday
9	Bryan	Thursday
10	Bryan	Friday
11	James	Monday
12	James	Tuesday
13	James	Wednesday
14	James	Thursday
15	James	Friday
16	Nicky	Monday
17	Nicky	Tuesday
18	Nicky	Wednesday
19	Nicky	Thursday
20	Nicky	Friday
21	Tom	Monday
22	Tom	Tuesday
23	Tom	Wednesday
24	Tom	Thursday
25	Tom	Friday

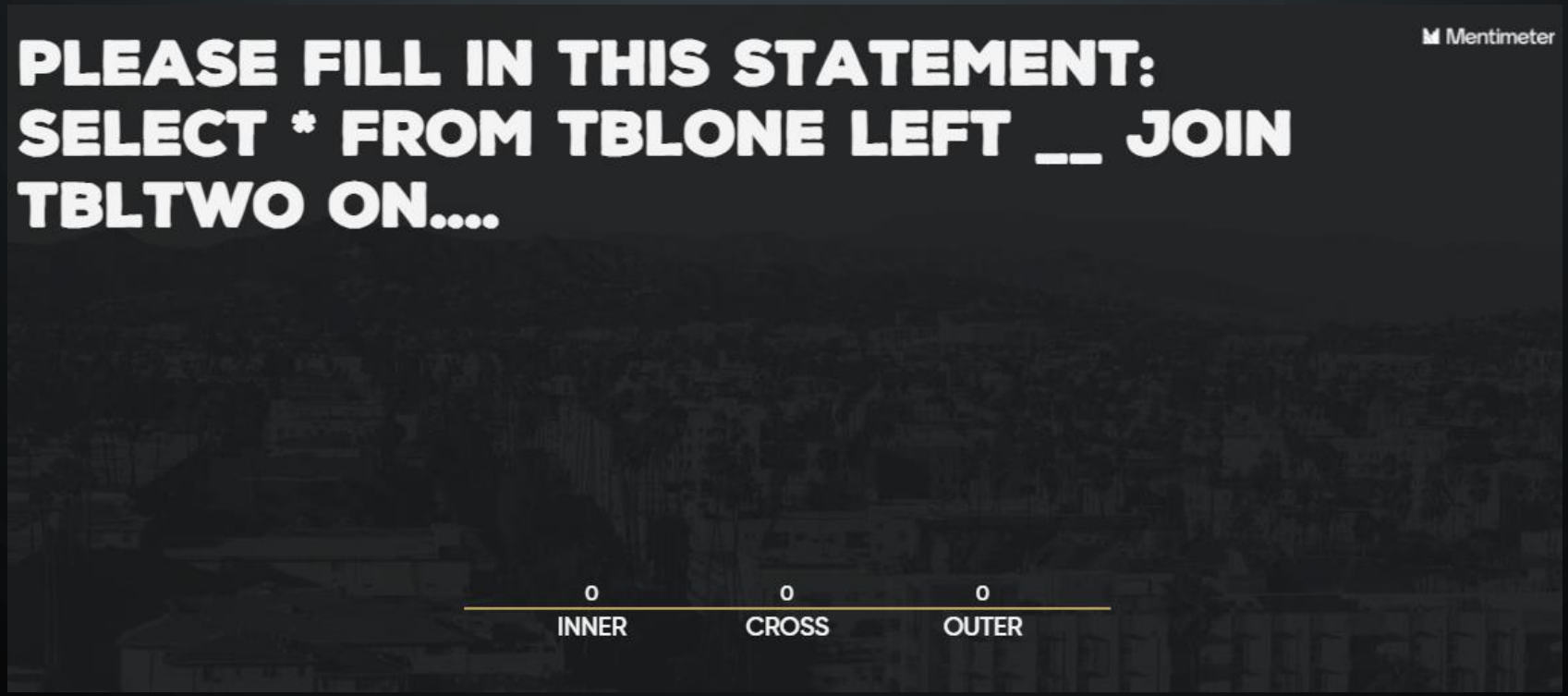
© Copyright 2021 Zenika. All rights reserved

QUESTION 6



© Copyright 2021 Zenika. All rights reserved

QUESTION 7



© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	**09**	**Set Operations**
Create Database and Table	04	10	Case statement
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

SET OPERATORS

Hm…Ok boss.. That's possible. Let me show you.

Marcus. Well done! Is it possible to combine the results of several queries?

© Copyright 2021 Zenika. All rights reserved

SET OPERATORS

- Results of multiple queries to be combined
- Set operators include UNION, UNION ALL, INTERSECT, and EXCEPT
- Queries must have same number of COLUMNS

© Copyright 2021 Zenika. All rights reserved

UNION

- UNION returns a result set of distinct rows combined from all statements
- UNION removes duplicates during query processing (affects performance)

© Copyright 2021 Zenika. All rights reserved

UNION

Continue….

```
-- It will take the column name FROM the first SELECT
SELECT 'hi' AS Greeting
UNION
SELECT 'hello there' AS GreetingNow;
```

Results | Messages

	Greeting
1	hi
2	hello there

```
-- UNION cannot have separate rows for same entry ('hi')
SELECT 'cat' AS Animal
UNION
SELECT 'turtle' AS Pet
UNION
SELECT 'cat'
UNION
SELECT 'snake';
```

Results | Messages

	Animal
1	cat
2	snake
3	turtle

© Copyright 2021 Zenika. All rights reserved

UNION ALL

Continue….

- UNION ALL retains duplicates during query processing

```
-- UNION ALL can have duplications
SELECT 'cat' AS Animal
UNION ALL
SELECT 'turtle' AS Pet
UNION ALL
SELECT 'cat'
UNION ALL
SELECT 'snake';
```

Results | Messages

	Animal
1	cat
2	turtle
3	cat
4	snake

```
select Location, LocationHead from tblLocation
union all
select TOP (10) CustomerFirstName, CustomerLastName from tblCustomer
```

Results | Messages

	Location	LocationHead
1	Central	Tom
2	North	James
3	South	Nicky
4	East	Andrew
5	West	Bryan
6	Jossef	Wright
7	Dylan	Word
8	Jossef	Wright
9	Jane	Zwilling
10	Carolyn	Zimmerman
11	Jane	Zabokritski
12	Ken	Yukish
13	Terri	Yu
14	Roberto	Young
15	Rob	Yalovsky

© Copyright 2021 Zenika. All rights reserved

EXCEPT

- EXCEPT returns only distinct rows that appear in the first set but not the second
- Order in which sets are specified matters

```
CREATE TABLE tblTemp1 (
    number int NOT NULL
);

INSERT INTO tblTemp1
VALUES (1), (2), (3), (4), (5);

CREATE TABLE tblTemp2 (
    number int NOT NULL
);

INSERT INTO tblTemp2
VALUES (1), (2), (2), (4), (5);
```

```
SELECT * from tblTemp1
EXCEPT
SELECT * from tblTemp2;
```

Results

	number
1	3

© Copyright 2021 Zenika. All rights reserved

INTERCEPT

- INTERSECT returns only distinct rows that appear in both result sets

```
CREATE TABLE tblTemp1 (
    number int NOT NULL
);

INSERT INTO tblTemp1
VALUES (1), (2), (3), (4), (5);

CREATE TABLE tblTemp2 (
    number int NOT NULL
);

INSERT INTO tblTemp2
VALUES (1), (2), (2), (4), (5);
```

```
SELECT * from tblTemp1
INTERSECT
SELECT * from tblTemp2;
```

Results

	number
1	1
2	2
3	4
4	5

© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	**10**	**Case statement**
Insert Data	05	11	Modifying Data
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

CASE STATEMENT

Good job Marcus! Some of our customers our very supportive. I want to identify and categorize them!

That's not too difficult!

© Copyright 2021 Zenika. All rights reserved

CASE

Continue….

- Evaluates a list of conditions and returns one of multiple possible result expressions.

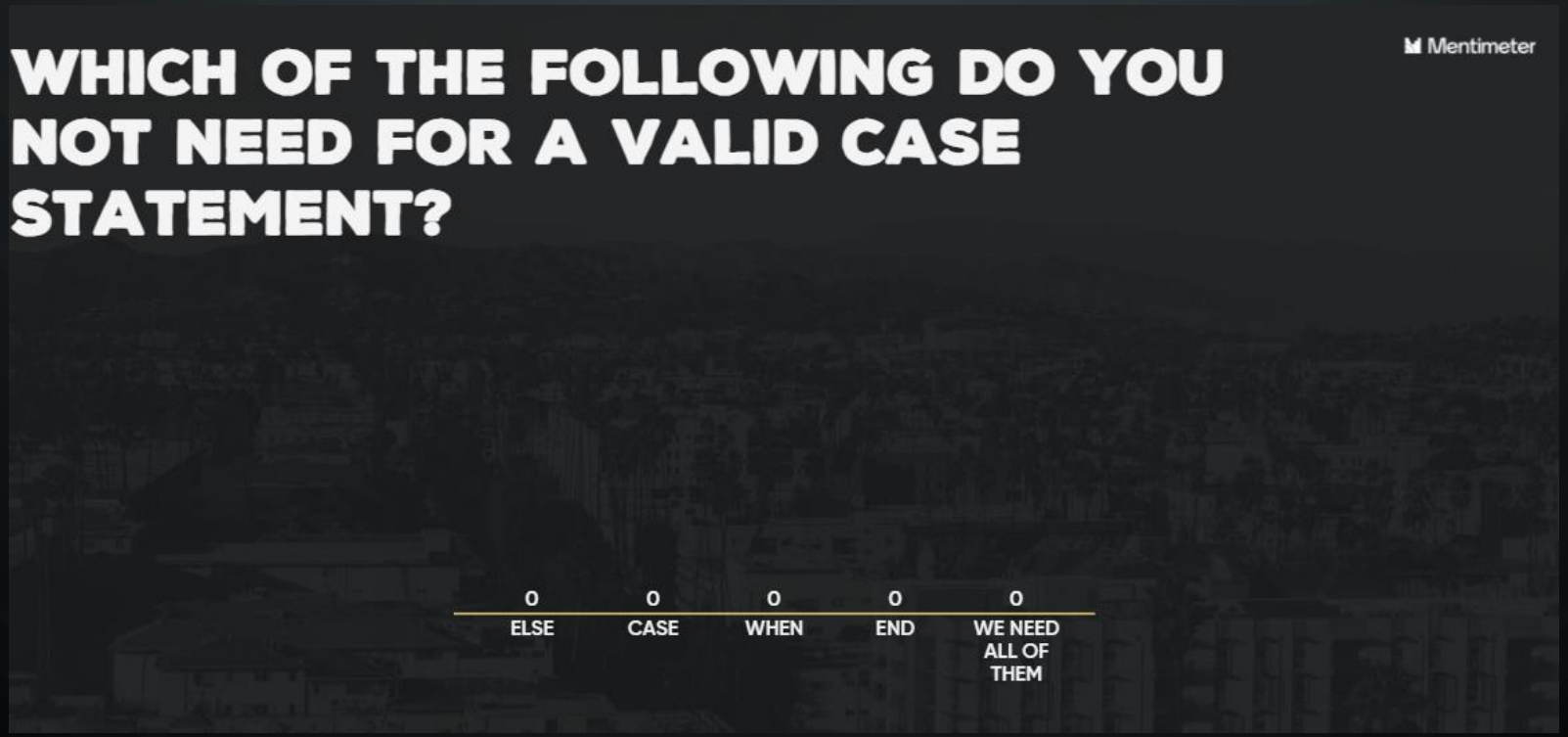
Total Amount of Transaction

>= 2000	→ Platinum
>= 1000	→ Gold
>= 500	→ Silver
< 500	→ Normal

```
SELECT
    CustomerNumber,
    COUNT(AMOUNT) AS NumberOfTransactions,
    SUM(AMOUNT) AS TotalAmount,
    CASE
        WHEN SUM(AMOUNT) >= 2000 THEN 'Platinum'
        WHEN SUM(AMOUNT) >= 1000 THEN 'Gold'
        WHEN SUM(AMOUNT) >= 500 THEN 'Silver'
        ELSE 'Normal'
    END AS Membership
FROM tblTransaction
GROUP BY CustomerNumber
ORDER BY CustomerNumber;
```

© Copyright 2021 Zenika. All rights reserved

QUESTION 8



© Copyright 2021 Zenika. All rights reserved

NEXT

Business intelligence (BI)	01	07	Aggregating Data
Database and RDBMS	02	08	Join Tables
SQL and T-SQL	03	09	Set Operations
Create Database and Table	04	10	Case statement
Insert Data	05	**11**	**Modifying Data**
Select Statement	06	12	Q&A

© Copyright 2021 Zenika. All rights reserved

UPDATE DATA

Well, if you say so…

Marcus, can you update the CustomerNumber of Customer # 10 to # 14 to 200? They are actually the same person.

© Copyright 2021 Zenika. All rights reserved

UPDATE DATA

Continue….

- Updates all rows in a table or view
- Set can be filtered with a WHERE clause
- Only columns specified in the SET clause are modified

```
UPDATE tblTransaction
SET CustomerNumber = 200
WHERE CustomerNumber in (10, 11, 12, 13, 14);
```

© Copyright 2021 Zenika. All rights reserved

DELETE DATA

Boss, we should have cleaned up the data earlier!

Sorry Marcus, I notice that some transactions belong to customers that do not exist. Can you delete them?

© Copyright 2021 Zenika. All rights reserved

DELETE DATA

Continue….

- DELETE deletes rows
- Use a WHERE clause to delete specific rows

```
-- Final execution
DELETE tblTransaction
FROM tblCustomer AS C
RIGHT JOIN tblTransaction T
    ON C.CustomerNumber = T.CustomerNumber
WHERE C.CustomerNumber IS NULL;
```

© Copyright 2021 Zenika. All rights reserved

DELETE AND TRUNCATE TABLE

.....
.....
.....

Marcus, you did an excellent job. Well, sorry to say that I gave you the wrong data from the beginning.
Can we restart?

© Copyright 2021 Zenika. All rights reserved

DELETE AND TRUNCATE TABLE

Method 1: Truncate all the data in the tables

```
TRUNCATE TABLE [tblLocation];

TRUNCATE TABLE [tblCustomer];

TRUNCATE TABLE [tblTransaction];
```

Method 2: Drop all tables

```
IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tblLocation]') AND type in (N'U'))
DROP TABLE [dbo].[tblLocation]
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tblCustomer]') AND type in (N'U'))
DROP TABLE [dbo].[tblCustomer]
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tblTransaction]') AND type in (N'U'))
DROP TABLE [dbo].[tblTransaction]
```

© Copyright 2021 Zenika. All rights reserved

NEXT

01 Business intelligence (BI)

02 Database and RDBMS

03 SQL and T-SQL

04 Create Database and Table

05 Insert Data

06 Select Statement

07 Aggregating Data

08 Join Tables

09 Set Operations

10 Case statement

11 Modifying Data

12 Q&A

© Copyright 2021 Zenika. All rights reserved

WHAT IS NEXT?

Sure boss!

Thanks for your effort! You did a great job! I am looking for more advanced SQL operations that you can do!

© Copyright 2021 Zenika. All rights reserved

CONCLUSION

- T-SQL can't be mastered in 2 hours.
- Take this as a beginning and explore the SQL world.
- It is an important skill set that you should dive into.
- Practice, practice and practice.
- Look into more advanced topics, such as View, Stored Procedures, Functions, Transaction, Triggers, Temporal table, Pivot, Indexes, Optimisation, etc..

© Copyright 2021 Zenika. All rights reserved

© Copyright 2021 Zenika. All rights reserved

FEEDBACK TIME…

© Copyright 2021 Zenika. All rights reserved

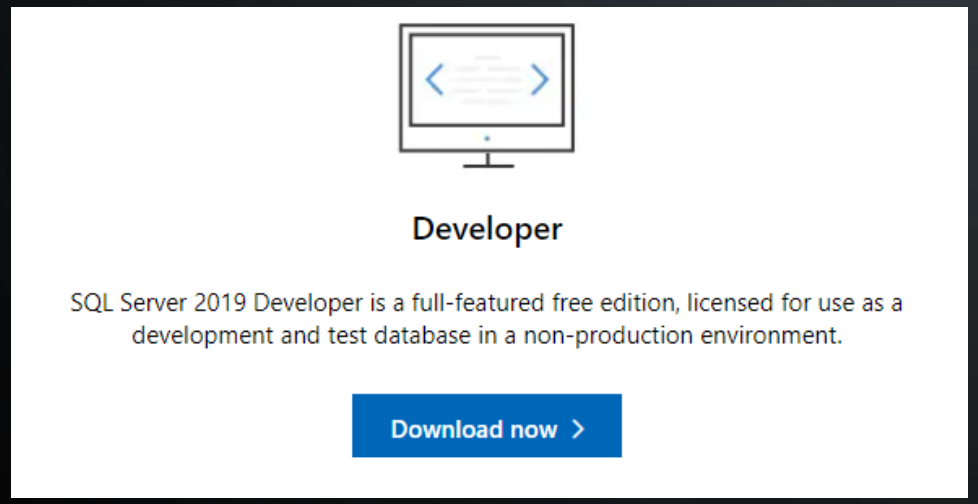
FEEDBACK TIME

© Copyright 2021 Zenika. All rights reserved

© Copyright 2021 Zenika. All rights reserved

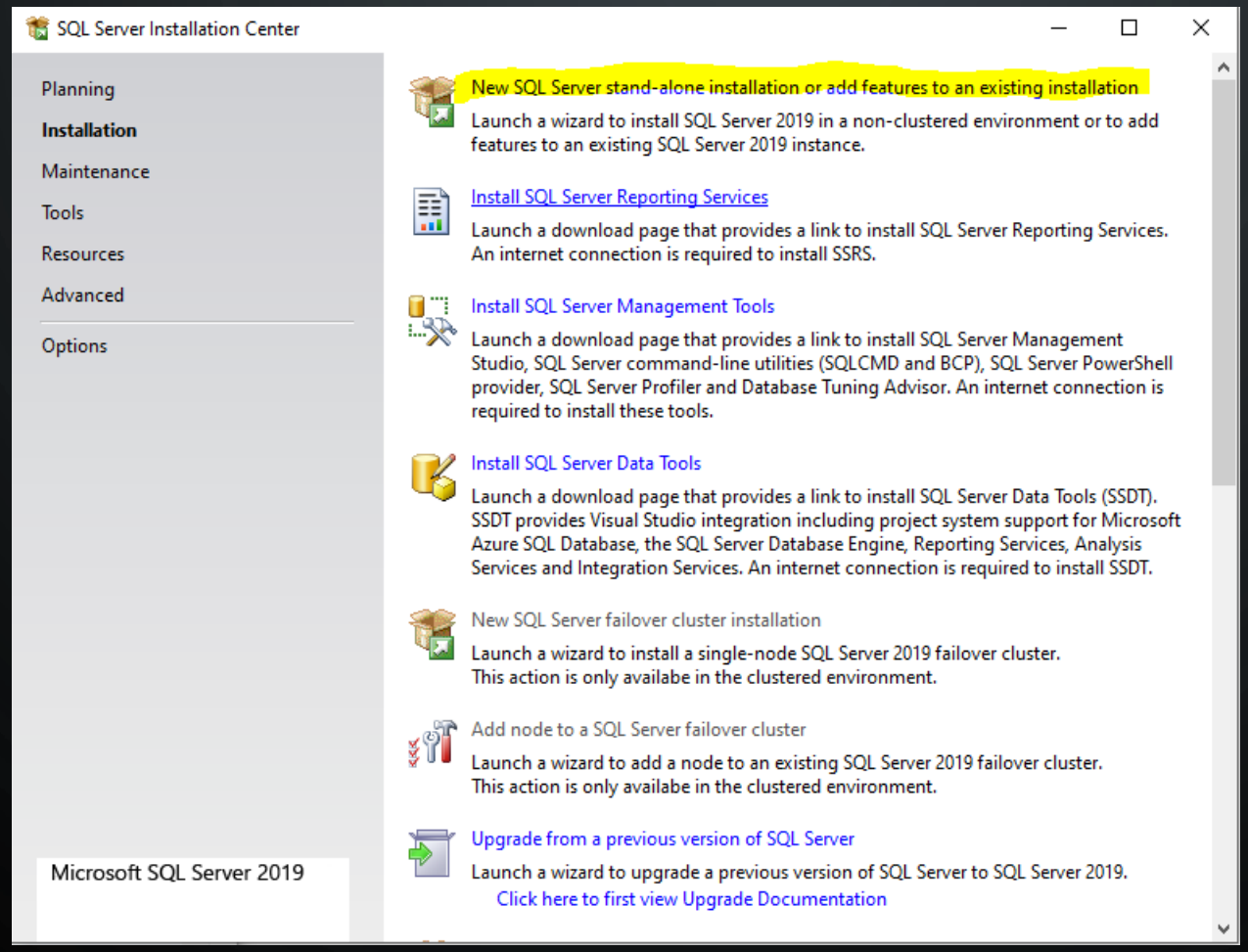
DOWNLOAD SQL SERVER 2019

- https://www.microsoft.com/en-gb/sql-server/sql-server-downloads



© Copyright 2021 Zenika. All rights reserved

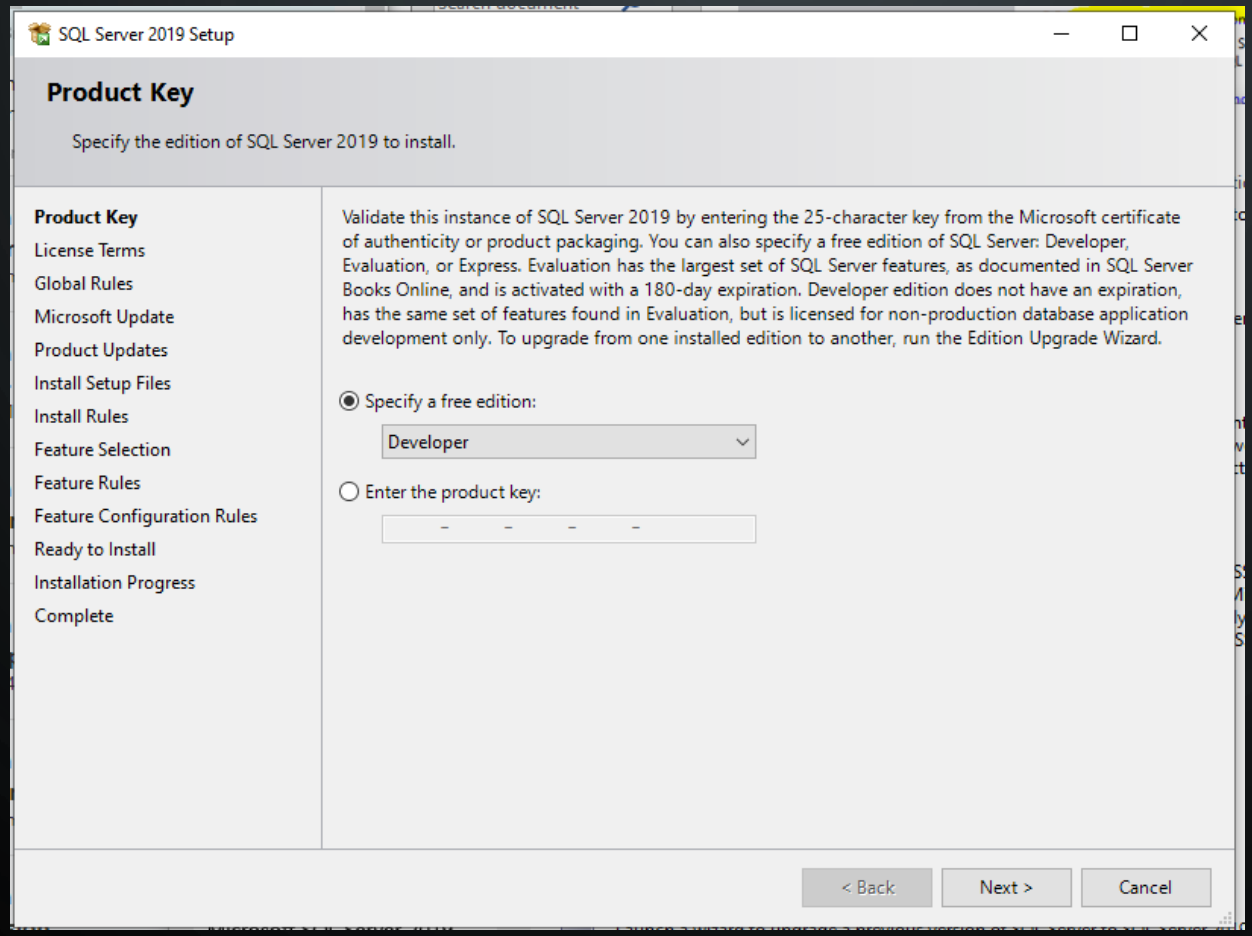
INSTALLING SQL SERVER DEVELOPER 2019



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

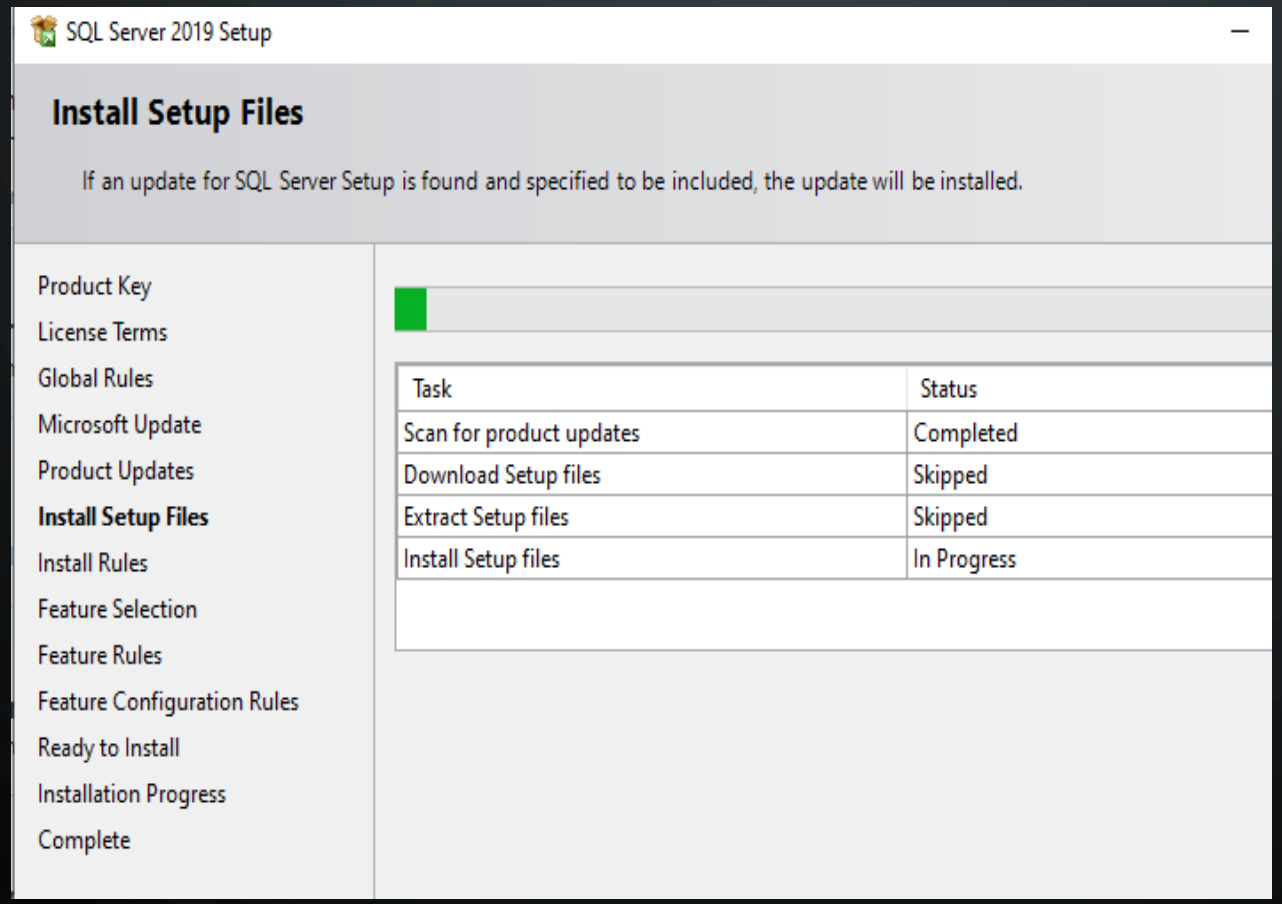
Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

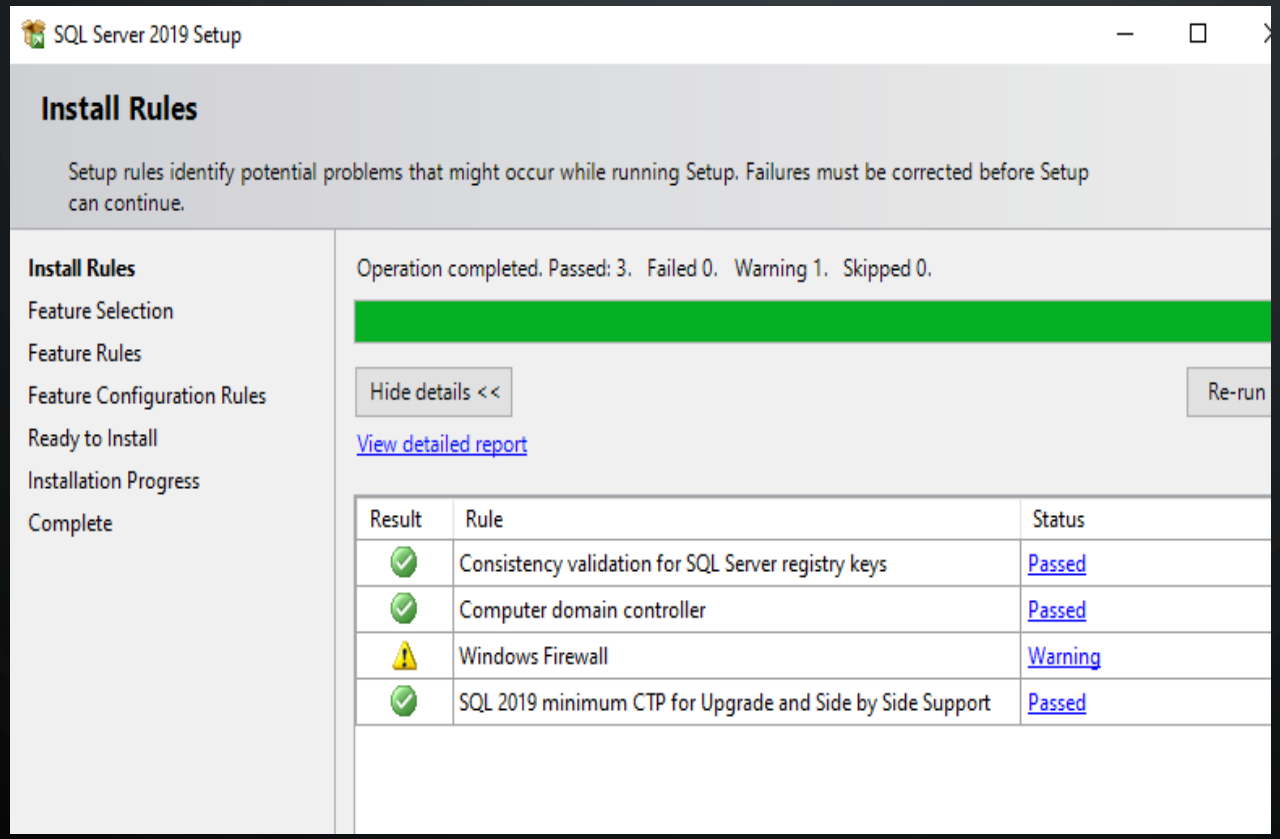
Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

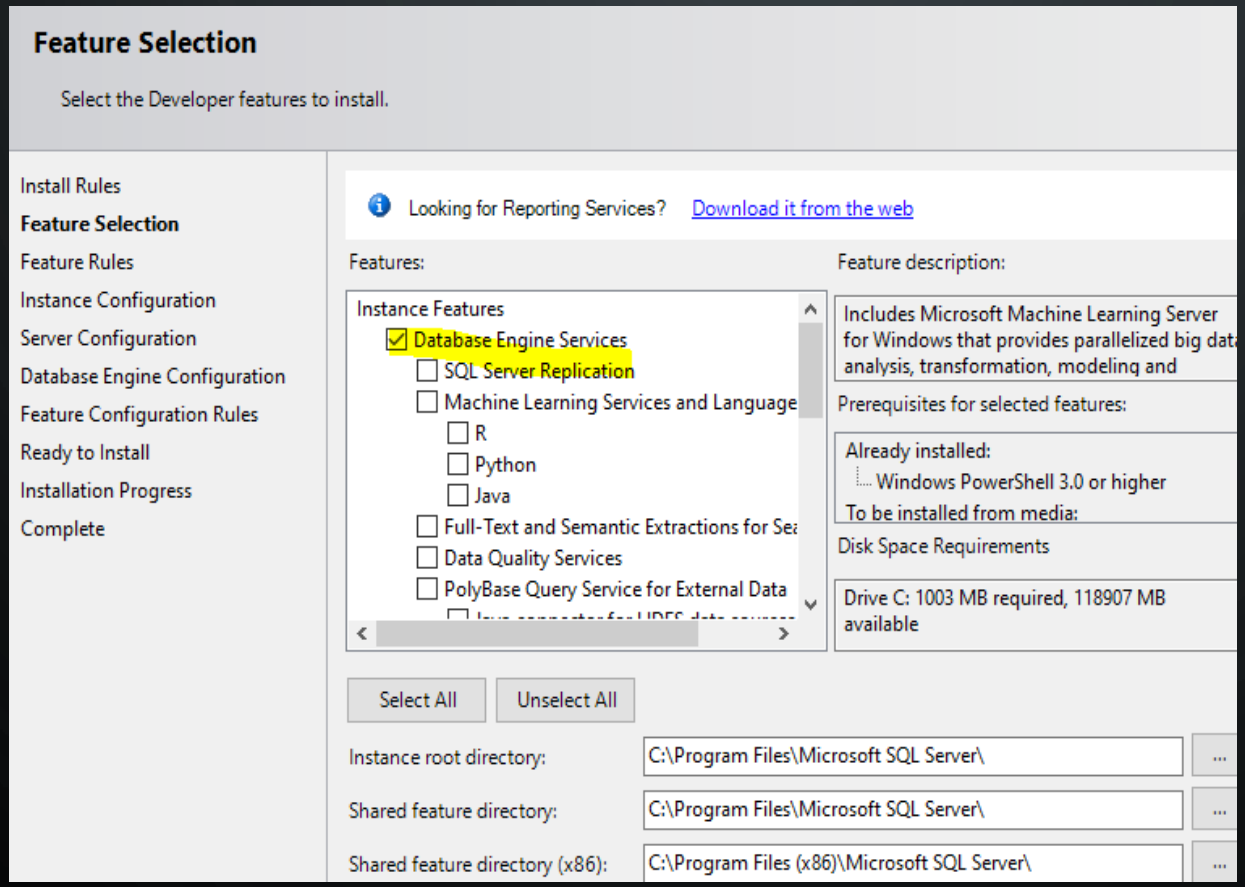
Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

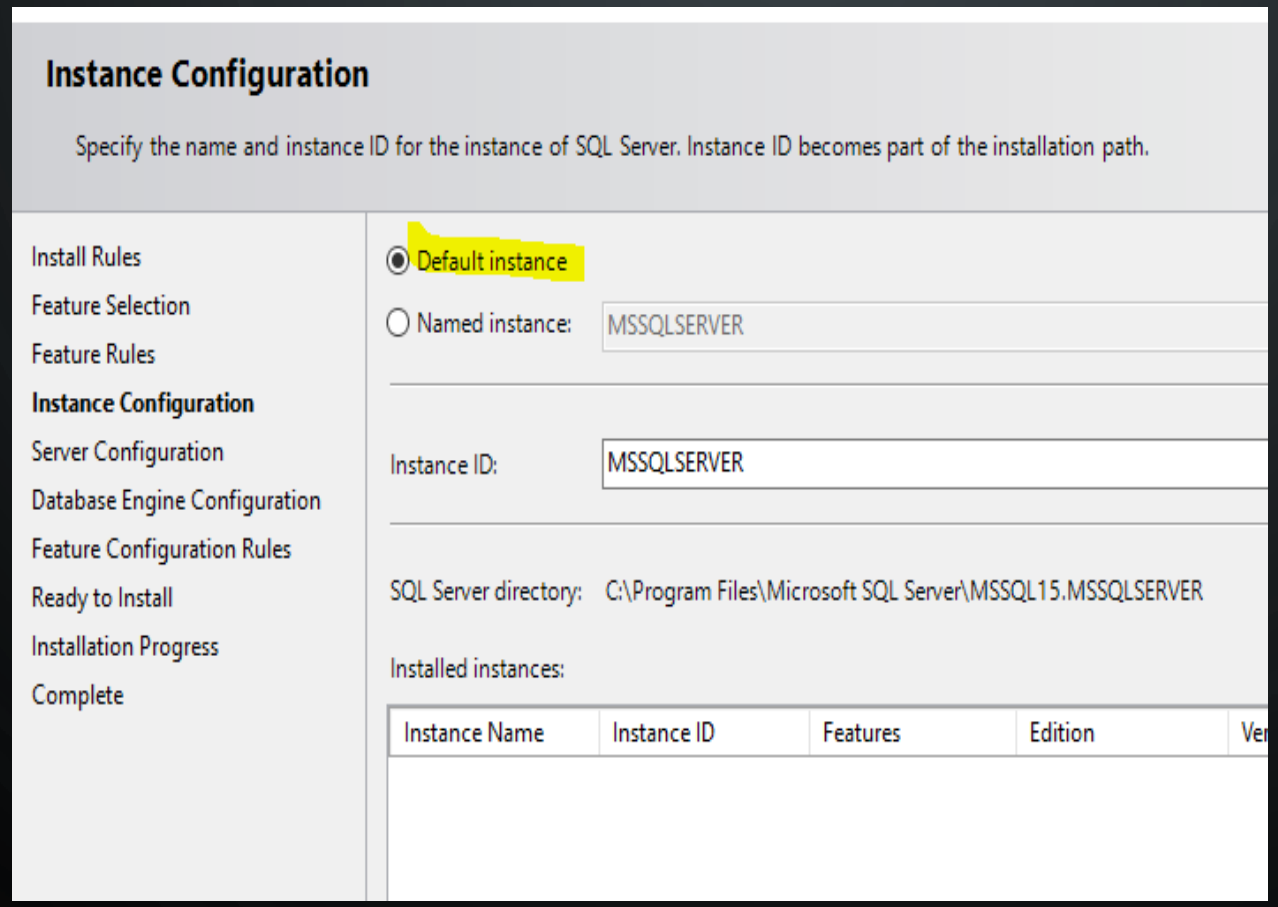
Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

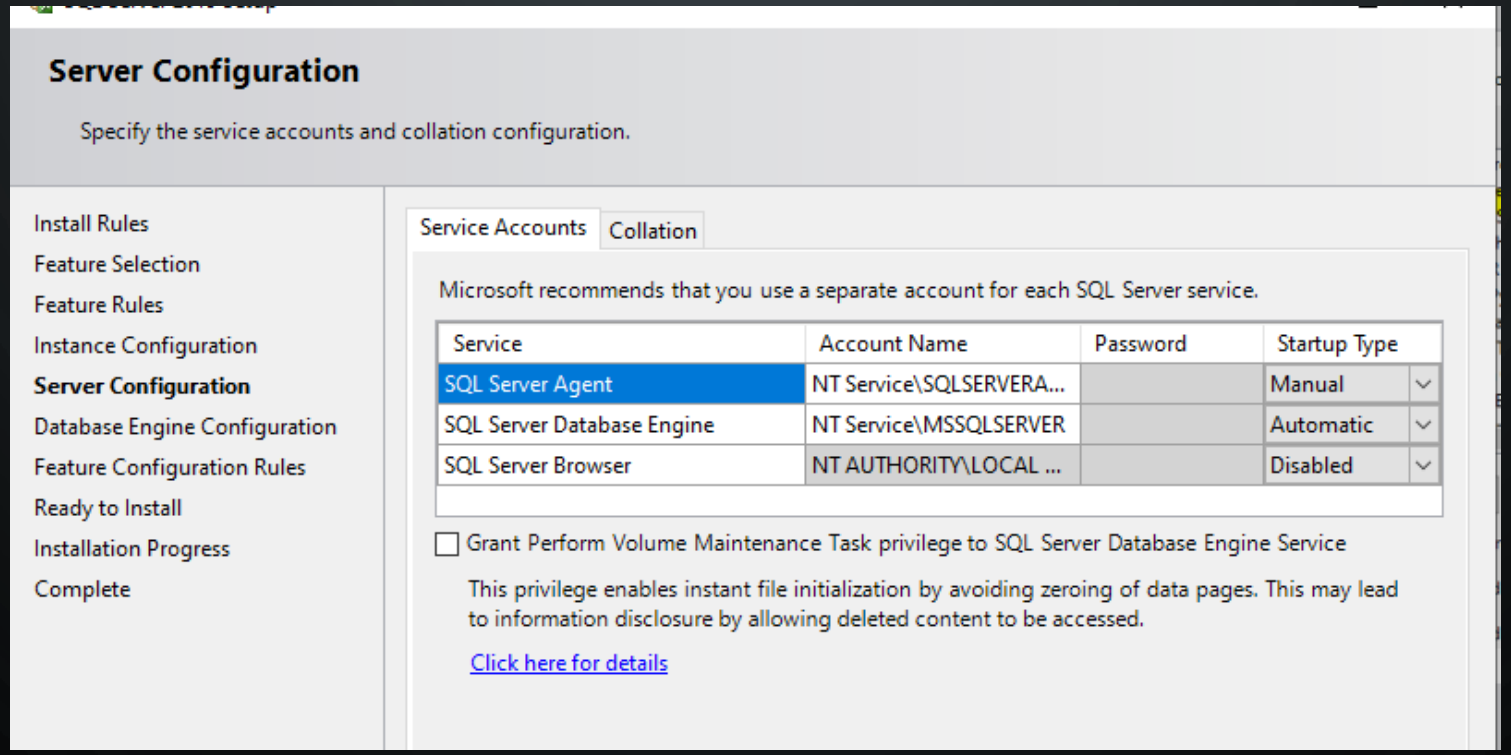
Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

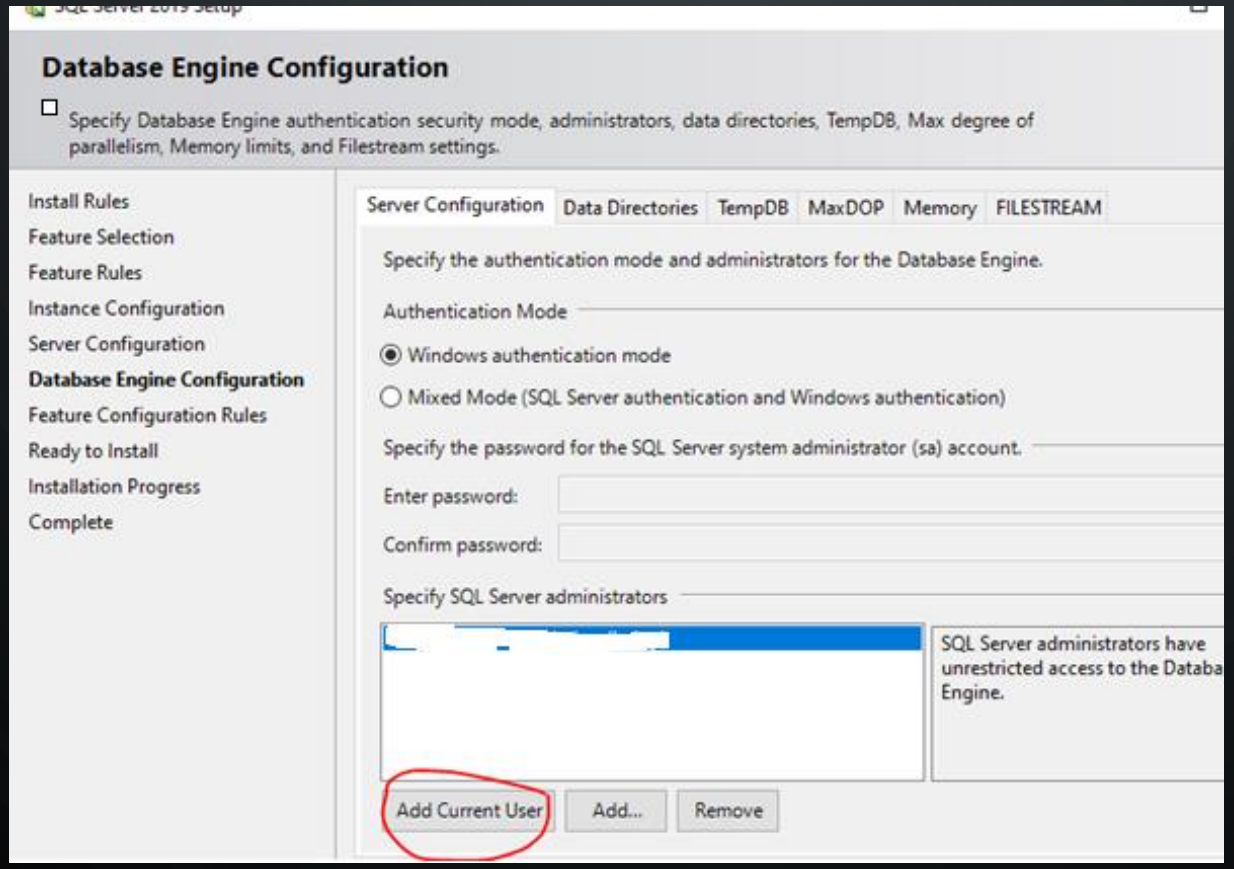
Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING SQL SERVER DEVELOPER 2019

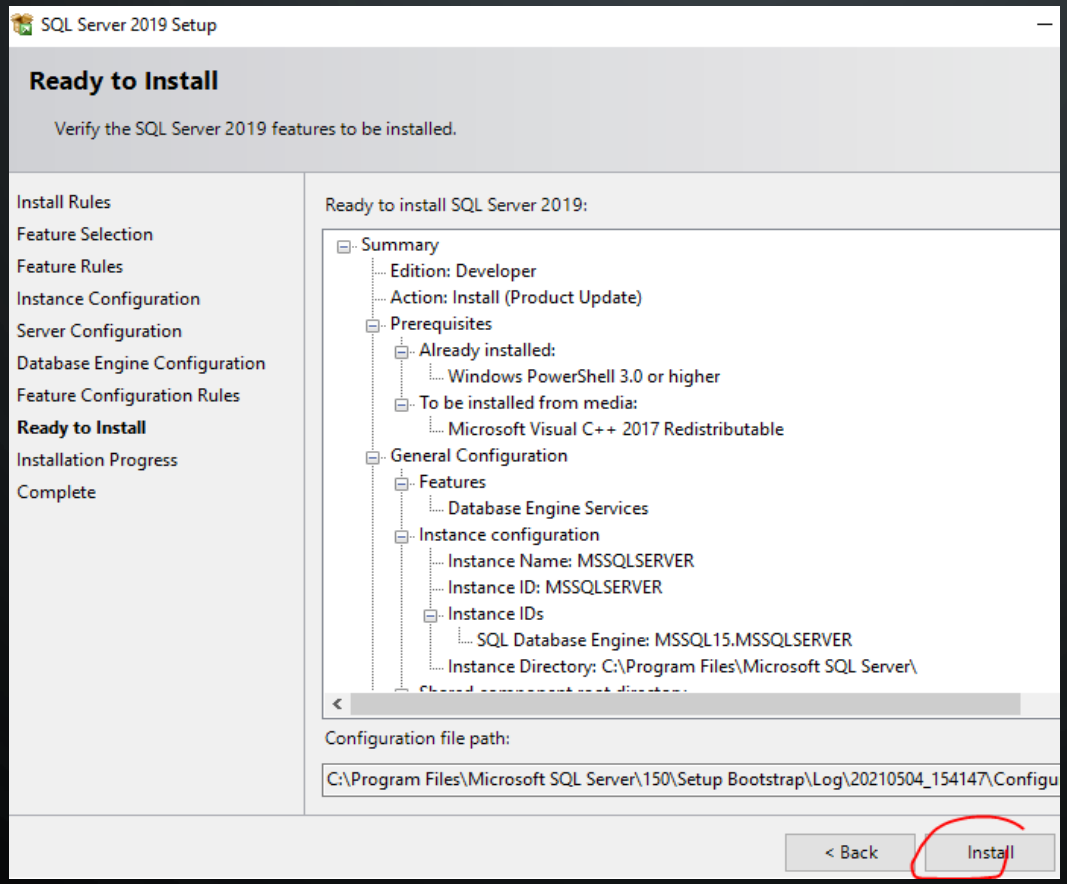
Continue….



© Copyright 2021 Zenika. All rights reserved

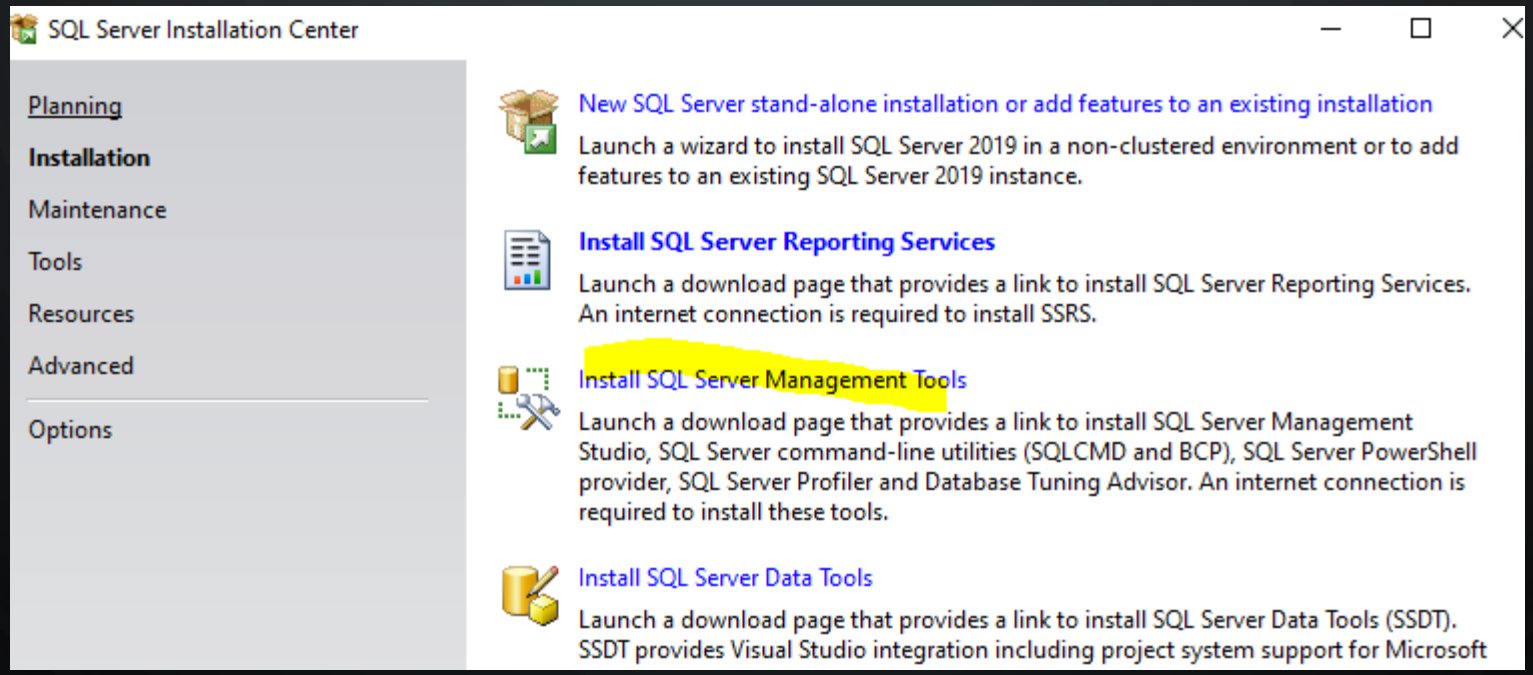
INSTALLING SQL SERVER DEVELOPER 2019

Continue….



© Copyright 2021 Zenika. All rights reserved

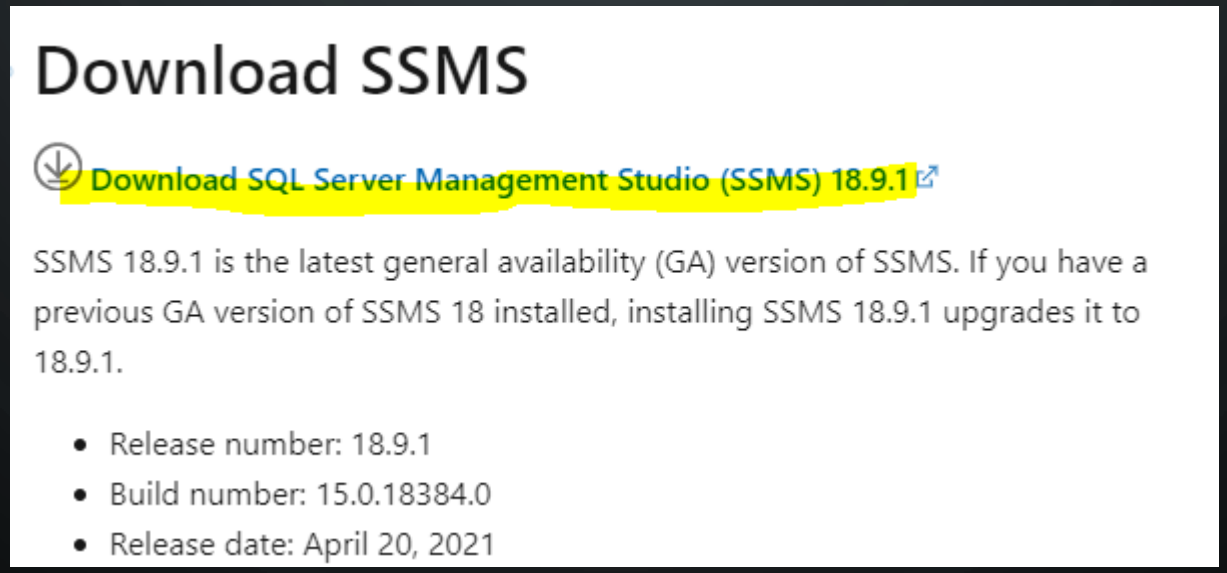
INSTALLING MICROSOFT SSMS



© Copyright 2021 Zenika. All rights reserved

INSTALLING MICROSOFT SSMS

Continue….



© Copyright 2021 Zenika. All rights reserved

INSTALLING MICROSOFT SSMS

Continue….