**Resume SQL SERVER**

**Dokumen ini Cuma buat review doang, bahasanya campur (indonesia, inggirs, sunda)**

1. SQL Server menggunakan transact SQL, Apa itu Transact SQL?

T-SQL (Transact-SQL) is a set of programming extensions from Sybase and Microsoft that add several features to the Structured Query Language ([SQL](http://searchsqlserver.techtarget.com/definition/SQL)) including transaction control, exception and error handling, row processing, and declared variables. Microsoft's [SQL Server](http://searchsqlserver.techtarget.com/definition/SQL-Server) and Sybase's SQL server support T-SQL statements.

1. Bagaimana arsitektur SQL SERVER?

General Architecture

**Client** − Where the request initiated.

**Query** − SQL query which is high level language.

**Logical Units** − Keywords, expressions and operators, etc.

**N/W Packets** − Network related code.

**Protocols** − In SQL Server we have 4 protocols.

* Shared memory (for local connections and troubleshooting purpose).
* Named pipes (for connections which are in LAN connectivity).
* TCP/IP (for connections which are in WAN connectivity).
* VIA-Virtual Interface Adapter (requires special hardware to set up by vendor and also deprecated from SQL 2012 version).

**Server** − Where SQL Services got installed and databases reside.

**Relational Engine** − This is where real execution will be done. It contains Query parser, Query optimizer and Query executor.

**Query Parser (Command Parser) and Compiler (Translator)** − This will check syntax of the query and it will convert the query to machine language.

(untuk lebih detail lagi tentang arsitektur sql server klik [disini...](%5bSQL%20SERVER%5d%20Architecture.docx))

1. SQL SYNTAX

Berikut beberapa syntax penting yang ada pada SQL

(http://www.w3schools.com/sql)

**SELECT** - extracts data from a database

**UPDATE** - updates data in a database

**DELETE** - deletes data from a database

**INSERT INTO** - inserts new data into a database

**CREATE DATABASE** - creates a new database

**ALTER DATABASE** - modifies a database

**CREATE TABLE** - creates a new table

**ALTER TABLE** - modifies a table

**DROP TABLE** - deletes a table

**CREATE INDEX** - creates an index (search key)

**DROP INDEX** - deletes an index

Berikut beberapa penggunaan Syntax/pengetahuan yang belum/ jarang maneh pake GIYAN!!

1. DISTINCT

SELECT DISTINCT City FROM Customers;

Men select data yang unik aja kalau dari kolom yang ada, kalau data nya ada yang sama (duplikasi) maka yang diambil Cuma satu.

1. SQL INJECTION (INI PENGETAHUAN AJA)

SQL INJECTION adalah sebuah istilah dimana hacker dapat memodifikasi sebuah query dengan menggunakan input pada kotak input yang disediakan pada sebuah website (e.g username & password) yang dapat mempengaruhi kode query pada server.

1. SELECT TOP

The SELECT TOP clause is used to specify the number of records to return. (inti na nge bates braha record nu rek diambil)

The SELECT TOP clause can be very useful on large tables with thousands of records. Returning a large number of records can impact on performance. **Note:** Not all database systems support the SELECT TOP clause.

SELECT TOP number|percent column\_name(s)  
FROM table\_name;

1. SQL WILDCARDS CHARACTER

In SQL, wildcard characters are used with the SQL LIKE operator.

SQL wildcards are used to search for data within a table.

Berikut beberapa karakter sql wildcards :

% A substitute for zero or more character

\_ A substitutr for a single character

[charlist] Sets and range of character to match

[^charlist] Matches only a character

[!charlist] NOT specified within the brackets

1. SQL IN

Perhatikeun syntax ieu :

SELECT \* FROM Customers

WHERE City IN ('Paris', 'London')

Syntax eta nu diluhur sarua jeung ieu yeuh ...

SELECT \* FROM Customers

WHERE City = 'Paris' OR City = 'London'

Nangkep maksud na? Sip ...

1. SQL BETWEEN

The following SQL statement selects all products with a price BETWEEN 10 and 20:

SELECT \* FROM Products  
WHERE Price BETWEEN 10 AND 20;

Untuk tipe data number udah dicoba dan bisa, untuk char/ text di contoh bisa tapi pas di cobain di sql server ga bisa.

1. ALIASES
2. Column alias

SELECT column\_name AS alias\_name  
FROM table\_name;

1. Table alias

SELECT column\_name(s)  
FROM table\_nameAS alias\_name;

1. Contoh

SELECT o.OrderID, o.OrderDate, c.CustomerName  
FROM Customers AS c, Orders AS o  
WHERE c.CustomerName="Around the Horn" AND c.CustomerID=o.CustomerID;

1. SQL JOIN
2. JOIN (INNER JOIN <- defaultnya)

SELECT column\_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column\_name=table2.column\_name;

1. LEFT JOIN/ LEFT OUTER JOIN (pada sebagian database)

The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

Jadi intinya kalau kondisi match nya ga terpenuhi maka data dari table1 (left table) tetep bakalan di tampilin

SELECT column\_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column\_name=table2.column\_name;

1. RIGHT JOIN

Sama kayak left join Cuma ini mah right nya.

1. FULL OUTER JOIN

Sama kayak left join, Cuma apabila si kondisi nya ga match maka nilai dari table1 dan table2 nya dua2 nya tetep bakalan di tampilin

SELECT column\_name(s)  
FROM table1  
FULL OUTER JOIN table2  
ON table1.column\_name=table2.column\_name;

1. UNION

SELECT column\_name(s) FROM table1  
UNION  
SELECT column\_name(s) FROM table2;

Si column\_name(s) nya harus sama antara tabel 1 dan 2

Secara default union langsugn pake distinct (milih data nya Cuma satu apabila data tersebut ada yang sama(duplikasi)) kalau mau dipilih semua pake UNION ALL

1. SELECT INTO

SELECT \*  
INTO newtable [IN externaldb] //nu di [] opsional jang db lain  
FROM table1;

Contoh :

SELECT \*  
INTO CustomersBackup2013  
FROM Customers;

1. INSERT INTO SELECT

The INSERT INTO SELECT statement copies data from one table and inserts it into an existing table. Jadi inti na select value ti table lain terus ngopikeun/ insert value eta kana table nu geus aya (lain tabel anyar) dengan sarat tipe data kolom na sarua

INSERT INTO table2  
SELECT \* FROM table1;

1. SQL CONSTRAINTS

CREATE TABLE table\_name  
(  
column\_name1 data\_type(size) constraint\_name,  
column\_name2 data\_type(size) constraint\_name,  
column\_name3 data\_type(size) constraint\_name,  
....  
);

* **NOT NULL** - Indicates that a column cannot store NULL value
* **UNIQUE** - Ensures that each row for a column must have a unique value
* **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Ensures that a column (or combination of two or more columns) have a unique identity which helps to find a particular record in a table more easily and quickly
* **FOREIGN KEY** - Ensure the referential integrity of the data in one table to match values in another table
* **CHECK** - Ensures that the value in a column meets a specific condition

CREATE TABLE Persons  
(  
P\_Id int NOT NULL CHECK (P\_Id>0),  
LastName varchar(255) NOT NULL,  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255)  
)

* **DEFAULT** - Specifies a default value for a column

CREATE TABLE Orders  
(  
O\_Id int NOT NULL,  
OrderNo int NOT NULL,  
P\_Id int,  
OrderDate date DEFAULT GETDATE()  
)

1. INDEX

The CREATE INDEX statement is used to create indexes in tables.

Indexes allow the database application to find data fast; without reading the whole table.

An index can be created in a table to find data more quickly and efficiently.

The users cannot see the indexes, they are just used to speed up searches/queries.

**Note:** Updating a table with indexes takes more time than updating a table without (because the indexes also need an update). So you should only create indexes on columns (and tables) that will be frequently searched against.

Add index :

CREATE INDEX PIndex  
ON Persons (LastName)

Drop index :

DROP INDEX table\_name.index\_name (SQL SERVER)

1. ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

Add column :

ALTER TABLE table\_name  
ADD column\_name datatype

Drop column :

ALTER TABLE table\_name  
DROP COLUMN column\_name

Modify column (SQL SERVER) :

ALTER TABLE table\_name  
ALTER COLUMN column\_name datatype

Add constraint sql server :

ALTER TABLE table\_name1 ADD CONSTRAINT nama\_constraint FOREIGN KEY (nama\_id\_foreign\_key) REFERENCES tabel\_name2

1. VIEWS

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

Create view :

CREATE VIEW view\_name AS  
SELECT column\_name(s)  
FROM table\_name  
WHERE condition

Create or Replace VIEW :

CREATE OR REPLACE VIEW view\_name AS  
SELECT column\_name(s)  
FROM table\_name  
WHERE condition

Drop view :

DROP VIEW view\_name

1. IS NULL & IS NOT NULL

Contoh :

SELECT LastName,FirstName,Address FROM Persons  
WHERE Address IS NOT NULL

**SQL FUNCTION**

Oke, sekarang kita masuk ke sesi function di SQL.

Jadi secara garis besar ada dua jenis funciton di SQL, yaitu :

SQL Aggregate Functions :

SQL aggregate functions return a single value, calculated from values in a column.

Useful aggregate functions:

* **AVG() - Returns the average value**

SELECT AVG(column\_name) FROM table\_name

e.g :

SELECT ProductName, Price FROM Products  
WHERE Price>(SELECT AVG(Price) FROM Products);

* **COUNT() - Returns the number of rows**

SELECT COUNT(column\_name) FROM table\_name;

a. SQL COUNT(\*) syntax (returns the number of records in a table) :

SELECT COUNT(\*) FROM table\_name;

* 1. SQL COUNT(DISTINCT column\_name) syntax (returns the number of distinct values of the specified column) :

SELECT COUNT(DISTINCT column\_name) FROM table\_name;

e.g :

SELECT COUNT(CustomerID) AS OrdersFromCustomerID7 FROM Orders  
WHERE CustomerID=7;

* **FIRST() - Returns the first value**

Ini Cuma support di ms Acces doang (sql server pake top aja)

* **LAST() - Returns the last value**

Ini juga Cuma support di ms Access doang

* **MAX() - Returns the largest value**

SELECT MAX(column\_name) FROM table\_name;

e.g : SELECT MAX(Price) AS HighestPrice FROM Products;

* **MIN() - Returns the smallest value**

SELECT MIN(column\_name) FROM table\_name;

SELECT MIN(Price) AS LowestPrice FROM Products;

* **SUM() - Returns the** **sum**

SELECT SUM(column\_name) FROM table\_name;

e.g : SELECT SUM(Quantity) AS TotalItemsOrdered FROM OrderDetails;

* **GROUP BY()**

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

Note : Aggregate functions often need an added GROUP BY statement.

SELECT column\_name, aggregate\_function(column\_name)  
FROM table\_name  
WHERE column\_name operator value  
GROUP BY column\_name;

SELECT Shippers.ShipperName,COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders  
LEFT JOIN Shippers  
ON Orders.ShipperID=Shippers.ShipperID  
GROUP BY ShipperName

Jadi intina, ngekueri na hiji2 per kolom(Shippername) jadi teu langsung di sekaligus ken.

* **HAVING**

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

Jadi intina, having teh sarua kos where tapi khusus jang kondisi nu melibatken fungsi aggregate

SELECT column\_name, aggregate\_function(column\_name)  
FROM table\_name  
WHERE column\_name operator value  
GROUP BY column\_name  
HAVING aggregate\_function(column\_name) operator value;

e.g :

SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM (Orders  
INNER JOIN Employees  
ON Orders.EmployeeID=Employees.EmployeeID)  
GROUP BY LastName  
HAVING COUNT(Orders.OrderID) > 10;

SQL SCALAR FUNCTION

SQL scalar functions return a single value, based on the input value.

Useful scalar functions:

* **UCASE() - Converts a field to upper case**

SELECT UPPER(column\_name) FROM table\_name;

* **LCASE() - Converts a field to lower case**

SELECT LOWER(column\_name) FROM table\_name;

* **MID() - Extract characters from a text field**

SELECT MID(column\_name,start,length) AS some\_name FROM table\_name;

* **LEN() - Returns the length of a text field**

SELECT LEN(column\_name) FROM table\_name;

* **ROUND() - Rounds a numeric field to the number of decimals specified**

SELECT ROUND(column\_name,decimals) FROM table\_name;

* **NOW() - Returns the current system date and time**

SELECT NOW() FROM table\_name;

* **FORMAT() - Formats how a field is to be displayed**

SELECT FORMAT(column\_name,format) FROM table\_name;

STORED PROCEDURE

Oke jadi sekarang kita masuk ke T-SQL salah satu fitur nya yaitu Stored Procedure. Otong : Jadi apa stored procedure itu bang? Ane: pertanyaan bagus tong, jadi stored procedure itu =

A stored procedure is a group of sql statements that has been created and stored in the database. Stored procedure will accept input parameters so that a single procedure can be used over the network by several clients using different input data. Stored procedure will reduce network traffic and increase the performance. If we modify stored procedure all the clients will get the updated stored procedure

Coontoh penggunaan nya ini tong :

/\*

Stored Procedure GetstudentnameInOutputVariable is modified to collect the

email address of the student with the help of the Alert Keyword

\*/

CREATE PROCEDURE GetstudentnameInOutputVariable

(

@studentid INT, --Input parameter , Studentid of the student

@studentname VARCHAR (200) OUT, -- Output parameter to collect the student name

@StudentEmail VARCHAR (200)OUT -- Output Parameter to collect the student email

)

AS

BEGIN

SELECT @studentname= Firstname+' '+Lastname,

@StudentEmail=email FROM tbl\_Students WHERE studentid=@studentid

END

**Perbedaan Stored Procedure sama User Define Function**

+---------------------------------+----------------------------------------+

| Stored Procedure (SP) | Function (UDF - User Defined |

| | Function) |

+---------------------------------+----------------------------------------+

| SP can return zero , single or | Function must return a single value |

| multiple values. | (which may be a scalar or a table). |

+---------------------------------+----------------------------------------+

| We can use transaction in SP. | We can't use transaction in UDF. |

+---------------------------------+----------------------------------------+

| SP can have input/output | Only input parameter. |

| parameter. | |

+---------------------------------+----------------------------------------+

| We can call function from SP. | We can't call SP from function. |

+---------------------------------+----------------------------------------+

| We can't use SP in SELECT/ | We can use UDF in SELECT/ WHERE/ |

| WHERE/ HAVING statement. | HAVING statement. |

+---------------------------------+----------------------------------------+

| We can use exception handling | We can't use Try-Catch block in UDF. |

| using Try-Catch block in SP. | |

+---------------------------------+----------------------------------------+

1. SQL SUBQUERIES

Jadi gini, misal ada dua tabel salesmen sama order trus kita pengen nge get semua order yang dimiliki oleh salesmen yang berasal dari london. Maka query nya adalah:

1. **SELECT** \*
2. **FROM** orders
3. **WHERE** salesman\_id =
4. (**SELECT** salesman\_id
5. **FROM** salesman
6. **WHERE** city='London');

Contoh 2:

Query dibawah ini pengen milih customer yang memiliki grade yang berbeda dengan customer dari kota dallas

1. **SELECT** \*
2. **FROM** customer
3. **WHERE** NOT grade = ANY
4. (**SELECT** grade
5. **FROM** customer
6. **WHERE** city='Dallas');