**NAMA:SAPITRI**

**NIM:200250502059**

**JURUSAN:TEKNIK INFORMATIKA B**

**MATAKULIAH:SISTEM BASIS DATA**

**DOSEN:MUSLIADI KH,S,KOM.,M.KOM**

Penjelasan dan contoh

1.SQL HOME

SQL pengantar adalah Bahasa standar yang untuk mengakses dan memanipulasi database.

***CONTOH:***

***Select \* From Consumer***

2.SQL INTRO

SQL adalah bahasa standar untuk mengakses dan memanipulasi database.

3.SQL SYNTAX

SQL Syntax merupakan Databaseyang paling sering berisi satu atau lebih tabel. Pada setiapakandiidentifikasi dengan nama (misalnya. "penjual" atau "Pesanan"). Tabel berisi record (baris) dengan data.

4.SQL SELECT

SQL SELECT yaitu memilih data dari database dan juga dugunakan untuk mengambil data data dari tabel database yang mengembalikan data ini dalam bentuk tabel hasil.

***CONTOH: SELECT*column1*,* column2,  *FROM* table\_name**:

5.SQL SELECT DISTINCT

Perintah select distinct yang digunakan dalam menampilkan sebuah nilai yang berbeda. Pada sebuah table dan kolom yang,sering berisi banyak nilai duplikat; Dan juga terkadang kita hanya ingin membuat sebuah daftar nilai yang terlihat berbeda. Perintah select distinc yang digunakan untuk menampilkan pada nilai yang berbeda dari suatu data.

***CONTOH:***

***SELECTDISTINCT*column1*,* column2, ...**

***FROM*table\_name*;***

6.SQL WHERESQ AND,OR,NOT

* logika And merupakan sebuah operator yang dapat digunakan untuk menampilkan 2 kondisi atau lebih dari 2 kondisi
* Logika OR merupakan sebuah operator yang dapat digunakan untuk menentukan sebuah logika dalam satu kondisi yang apabilah dibandingkan dengan 2 kondisi dan yang satu salahnya maka OR akan menampilkan sebuah nilai yang benar.
* Logika NOT merupakan logika yang berfungsi sebagai logika yang tidak dapat kita kehendaki.

***CONTOH AND:***

***SELECT*column1*,* column2, ...**

***FROM*table\_name**

***WHERE*condition1*AND*condition2*AND*condition3 ...*;***

***CONTOH OR:***

***SELECT*column1*,* column2, ...**

***FROM*table\_name *WHERE*ondition1*OR*condition2*OR*condition3 ...*;***

***CONTOH NOT:***

***SELECT*column1*,* column2, ...**

***FROM*table\_name *WHERENOT*condition*;***

7.SQL ORDER BY

Order By yang dapat digunakan untuk mengurutkan sebuah hasil dalam urutan naik atau turun. Kata kunci pada order by yang mengurutkan rekaman pada sebuah urutan menaik secara default. Untuk mengurutkan sebuah rekaman dalam urutan menurun, dapat gunakan kata kunci DESC.

***CONTOH:***

***SELECT*column1*,*column2, ... *FROM*table\_name *ORDERBY*column1, column2, ... *ASC|DESC;***

8.SQL INSERT INTO

SQL Insert Into merupakan Pernyataan yang digunakan untuk memasukkan record baru ke dalam table.

***CONTOH:***

***INSERTINTO*table\_name *(*column1*,* column2*,*column3*, ...)  
VALUES (*value1*,* value2*,* value3*, ...);***

9.SQ NULL VALUES

SQL Update dapat digunakan untuk mengubah sebuah catatan atau isi yang terdapat pada data dalam sebuah table. Hal yang perlu diperhatikan dalam mempebaharui record pada sebuah table yaitu klausa WHERE karena jika kita menghilangkan klausa WHERE maka seluruh data pada table juga akan ikut berubah.

***CONTOH:***

***UPDATE*table\_name *SET*column1 *=* value1*,* column2 *=*value2*, ...  
WHERE*condition**

10.SQL UPDATE SQL DELETE

Pernyataan DELETE yang digunakan untuk menghapus rekaman yang terdapat dalam tabel.

***CONTOH:***

***DELETEFROM Customers WHERE CustomerName='Alfreds Futterkiste';***

11.SQL SELECT TOP

Select Top dapat digunakan untuk menentukan sebuah jumlah record yang akan dikembalikan. Klausa select top juga berguna pada tabel besar dengan ribuan record. Yang dapat mengembalikan sejumlah besar rekaman yang dapat memengaruhi kinerja.

***CONTOH***

***SELECTTOP*number*|percent*column\_name(s)**

***FROM*table\_name**

***WHERE*condition*;***

12.SQL MIN AND MAX

MIN dapat digunakan untuk mengembalikan nilai terkecil dari kolom yang dipilih. dan Max dapat digunakan untuk mengembalikan nilai terbesar dari kolom yang dipilih.

***CONTOH :***

***MIN() Syntax***

***SELECT MIN(column\_name)***

***FROM table\_name***

***WHERE condition;***

***MAX() Syntax***

***SELECT MAX(column\_name)  
FROM table\_name  
WHERE condition;***

13.COUNT,AVG,SUM

count dapat digunakan untuk mengembalikan sebuh jumlah baris yang cocok pada kriteria yang ditentukan, Avg yang berguna untuk mengembalikan nilai rata-rata pada kolom numerik. Sum dapat digunakan untuk mengembalikan jumlah total pada kolom numerik.

***CONTOH :***

***COUNT() Syntax***

***SELECT COUNT(column\_name)  
FROM table\_name  
WHERE condition;***

***AVG() Syntax***

***SELECT AVG(column\_name)  
FROM table\_name  
WHERE condition;***

***SUM() Syntax***

***SELECT SUM(column\_name)  
FROM table\_name  
WHERE condition;***

14.SQL LIKE

SQL Like biasa digunakan bersama-sama dengan perintah SELECT, dan juga biasanya dimanfaatkan dalam pencaarian data. **LIKE**artinya mirip. Oleh karena itu yang dicari adalah suatu data yang mirip dengan apa yang kita inginkan.

***CONNTOH:***

***SELECT*column1, column2, ... *FROM*table\_name *WHERE*columnN*LIKE*pattern*;***

15.SQL WILDCARDS

Wildcard merupakan sebuah karakter yang digunakan untuk menggantikan satu atau lebih karakter dalam sebuah string. Karakter wildcard dapat digunakan dengan operator [SQL LIKE](https://www.w3schools.com/sql/sql_like.asp)Operator LIKE yang dapat digunakan dalam klausa WHERE untuk mencari pola tertentu dalam kolom.

***CONTOH:***

***SELECT \* FROM Customers  
WHERE City LIKE'ber%';***

16.SQL IN

Operator IN dapat memungkinkan kita untuk menentukan beberapa nilai dalam sebuah klausa WHERE.Operator IN merupakan singkatan dari beberapa kondisi .

***CONTOH***

***SELECT*column\_name(s)**

***FROM*table\_name *WHERE*column\_name*IN (SELECT* STATEMENT*);***

17.SQL BATWEEN

Operator Between merupakan sebuah operator yang memilih nilai dalam rentang tertentu. Nilainya bisa berupa angka, teks, atau tanggal. Operator between bersifat inklusif, nilai awal dan akhir disertakan.

***CONTOH:***

***SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name BETWEEN value1 AND value2;***

18.SQL ALIASES

Alias ​​SQL dapat digunakan untuk memberi nama sementara pada sebuah tabel, atau kolom dalam tabel. Alias dan juga ​​sering digunakan untuk membuat nama kolom lebih mudah dibaca. Dan Alias ​​hanya ada selama kueri.

***CONTOH:  
Alias Column Syntax***

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

***Alias Table Syntax***

***SELECT column\_name(s)  
FROM table\_name AS alias\_name;***

20.SQL JOINS

Klausa JOIN dapat digunakan untuk menggabungkan sebuah baris dari dua atau lebih tabel, berdasarkan pada kolom terkait di antara mereka.

***CONTOH:***

***SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNERJOIN Customers ON Orders.CustomerID=Customers.CustomerID;***

21.SQL INNER JOIN

Pada Kata kunci Left Join dapat mengembalikan semua record dari table 1 kiri, dan record yang cocok dari table 2 kanan. Dan hasilnya adalah Null dari sisi kanan,apabilah tidak ada yang cocok.

***CONTOH:***

***SELECT*column\_name(s) *FROM*table1 *LEFTJOIN*table2 *ON*table1.column\_name *=* table2.column\_name*;***

22.SQL LEFT JOIN

Pada Kata kunci Left Join dapat mengembalikan semua record dari table 1 kiri, dan record yang cocok dari table 2 kanan. Dan hasilnya adalah Null dari sisi kanan,apabilah tidak ada yang cocok.

***CONTOH:***

***SELECT*column\_name(s) *FROM*table1 *LEFTJOIN*table2 *ON*table1.column\_name *=* table2.column\_name*;***

23.SQL RIGHT JOIN

Pada kata kunci Right Join dapat mengembalikan semua record dari tabel 2 pada bagian kanan ,dan record yang cocok untuk table 1 dari sisi kiri dan hasilnya adalah NULL dari sisi kiri, jika tidak ada yang cocok.

***CONTOH:***  
***SELECT*column\_name(s) *FROM*table1 *RIGHTJOIN*table2 *ON*table1.column\_name *=* table2.column\_name*;***

24.SQLFULL JOIN

Pada kata kunci Left Join dapat mengembalikan semua record dari table 1 bagian sisi kiri , dan record yang cocok dari table 2 pada sisi kanan . Dan hasilnya adalah Null dari sisi kanan, jika tidak ada yang cocok.

***CONTOH:***

***SELECT*column\_name(s) *FROM*table1 *FULLOUTERJOIN*table2 *ON*table1.column\_name *=* table2.column\_name *WHERE*condition*;***

25.SQL SELF JOIN

Pada kata kunci Left Join dapat mengembalikan semua record dari table 1 sisi kiri dan record yang cocok dari table 2 disisi kanan. Hasilnya adalah NULL dari sisi kanan, jika tidak ada yang cocok.

***CONTOH:***

***SELECT*column\_name(s) *FROM*table1 T1, table1 T2 *WHERE*condition*;***

26.SQL UNION

Pada Operator Union dapat digunakan untuk menggabungkan kumpulan-kumpulan dari hasil dua atau lebih pada pernyataan select. Pada setiap pernyataan select dalam union harus memiliki jumlah kolom yang sama, pada kolom juga harus memiliki tipe data yang serupa dan di setiap pernyataan select pada kolom juga harus dalam urutan yang sama.

***CONTOH:***

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

27.SQL GROUP BY

Pernyataan pada Group By dapat mengelompokkan baris yang memiliki nilai yang sama ke dalam sebuah baris ringkasan, seperti "temukan harga paada setiap barang". Pernyataan GROUP BY sering digunakan dengan fungsi agregat (COUNT, MAX, MIN, SUM, AVG) untuk mengelompokkan sebuah hasil yang ditetapkan oleh satu atau lebih dari sebuah kolom.

***CONTOH:***

***SELECT*column\_name(s) *FROM*table\_name *WHERE*condition *GROUPBY*column\_name(s) *ORDERBY*column\_name(s);**

28.SQL HAVING

Pada Klausa HAVING yang ditambahkan ke SQL karena kata kunci WHERE tidak dapat digunakan dengan fungsi agregat.

***CONTOH:***

***SELECT*column\_name(s) *FROM*table\_name *WHERE*condition *GROUPBY*column\_name(s) *HAVING*condition *ORDERBY*column\_name(s);**

29.SQL EXISTS

Operator EXISTS yaitu digunakan untuk menguji keberadaan rekaman apa pun pada sebuah subkueri. Operator EXISTS mengembalikan nilai trued dan jika subkueri mengembalikan satu atau lebih rekaman.

***CONTOH:  
SELECT*column\_name(s) *FROM*table\_name *WHEREEXISTS  
(SELECT*column\_name *FROM*table\_name*WHERE*condition*);***

30.SQL ANY,ALL

Pada Operator ANY dan ALL dapat digunakan dengan klausa WHERE atau HAVING. Operator ANY mengembalikan true jika salah satu dari nilai subkueri memenuhi kondisi. Operator ALL mengembalikan true jika semua nilai subkueri memenuhi kondisi.

***CONTOH:***

***ANY Syntax***

***SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name operator ANY  
(SELECT column\_name FROM table\_name WHERE condition);***

***ALL Syntax***

***SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name operator ALL  
(SELECT column\_name FROM table\_name WHERE condition);***

31.SQL SELECT INTO

Pada pernyataan Select Into dapat menyalin data dari satu tabel ke tabel baru.

***CONTOH:***

***SELECT \* INTO CustomersBackup2017  
FROM Customers;***

32.SQL INSERT INTO SELECT

Pada pernyataan Insert Into Select menyalin data dari satu tabel dan menyisipkannya ke tabel lain. Pada Insert Into Select mengharuskan jenis data pada tabel sumber dan target cocok, Catatan yang ada pada tabel target tidak terpengaruh.

**CONTOH:**

***INSERTINTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers;***

33.SQL CASE

pernyataan Case melewati kondisi dan mengembalikan nilai saat kondisi pertama terpenuhi (seperti pernyataan IF-THEN-ELSE). Jadi, apabilah suatu kondisi teah benar, maka itu berhenti membaca dan mengembalikan hasilnya. Apabilah tidak ada kondisi yang benar, ini mengembalikan nilai dalam klausa ELSE.

***CONTOH:***

***SELECT CustomerName, City, Country  
FROM Customers  
ORDERBY  
(CASE***

***WHEN City ISNULLTHEN Country***

***ELSE City  
END);***

34.SQL NULL FUNCTIONS

Pada bagian Null Functions yang merupakan fungsi Null yang didefenisikan oleh SQL dan pada SQL Saver funsi dari Isnnul digunakan untuk menggantikan nilai Null.

***CONTOH:***

***SELECT ProductName, UnitPrice \* (UnitsInStock + UnitsOnOrder)  
FROM Products;***

35.SQL STORED PROSEDURES

Prosedur yang tersimpan adalah kode SQL yang disiapkan yang dapat kita simpan, sehingga kode tersebut dapat digunakan kembali berulang kali. Jadi jika kita memiliki kueri SQL yang kita tulis berulang kali, maka simpanlah sebagai prosedur tersimpan, lalu panggil saja untuk menjalankannya. Kita juga dapat meneruskan parameter ke prosedur tersimpan, sehingga prosedur tersimpan dapat bertindak berdasarkan nilai parameter yang diteruskan.

***CONTOH:***

***CREATEPROCEDURE SelectAllCustomers  
AS  
SELECT \* FROM Customers  
GO;***

36.SQL COMMENTS

Comments yang digunakan untuk menjelaskan bagian-bagian dari pernyataan SQL, atau untuk mencegah eksekusi pernyataan SQL.

***CONTOH :***

***SELECT \* FROM Customers -- WHERE City='Berlin';***

37.SQL OPERATORS

Operators dalam SQL merupakan kata atau karakter yang sering digunakan terutama dalam Clause Where pada pernyataan SQL untuk melakukan sebuh operasi. 9ok,

38.SQL DATABASE

SQL Database merupakan tempat untuk menyimpan data kedalam sebuah table. Database relasional yang menyimpan dat kedalam sebua table .

***CONTOH:***

39.SQL CREATE DB

Pernyataan drop databasE digunakan untuk menjatuhkan database SQL yang ada.

***CONTOH:***

***DROPDATABASE testDB;***

40.SQL DROP DB

Pada pernyataan Backup Database dapat digunakan pada SQL Server untuk membuat sebuah cadangan penuh dari database SQL yang ada.

***CONTOH:***

***BACKUPDATABASE*databasename *TODISK = '*filepath*';***

41.SQL CREATE TABLE

Pada bagian pernyataan Create Table dapat kita digunakan untuk membuat sebuah tabel yang baru dalam sebuah database.

***CONTOH:***

***CREATETABLE TestTable AS  
SELECT customername, contactname  
FROM customers;***

42 SQL DROP TABLE

Pada bagian pernyataan Drop Table dapat kita gunakan untuk menjatuhkan suatu tabel yang ada dalam suatu database.

***CONTOH:***

***DROPTABLE Shippers;***

43.SQL ALTER TABLE

Alter Table merupakan fungsi atau Query SQL yang dapat kita gunakan untuk mengubah suatun struktur table pada suatu database. Pernyataan suatu Alter Table dapat kita gunakan untuk menambahkan, menghapus, atau memodifikasi suatu kolom dalam tabel yang telah ada.

***CONTOH:***

***ALTERTABLE Customers  
ADD Email varchar(255);***

44.SQL CONSTRAINSTS

Constrants merupakan suatu pembatasan yang dapat kita ditempatkan pada satu atau beberapa kolom tabel untuk membatasi jenis nilai yang dapat kita disimpan dalam sebuah kolom . Constraint juga menyediakan mekanisme standar untuk menjaga akurasi dan integritas data di dalam suatu tabel database

***CONTOH***

**CREATETABLEtable\_name (  
column1 datatypeconstraint,**

**column2 datatypeconstraint,**

**column3 datatypeconstraint,  
    ....  
);**

45.SQL NOT NULL

Pada Atribut Not Null dapat kita gunakan pada hampir semua jenis tipe data, Fungsinya yaitu untuk memastikan bahwa nilai pada suatu kolom tersebut tidak boleh dikosongkan. Jika kita ingin menginput data, namun tidak akan memberikan nilai untuk kolom tersebut, akan tetapi menghasilkan error pada MySQL.

***CONTOH:***

**CREATETABLE Persons (  
ID int NOTNULL,  
LastName varchar(255) NOTNULL,  
FirstName varchar(255) NOTNULL,  
Age int *);***

46.SQL UNIQUE

Unique mempunyai fungsi yaitu pada dasarnya sama seperti primary key, yaitu untuk memastikan bahwa pada setiap baris data yang terdapat dalam suatu tabel mempunyai sifat unik dengan kata laintidak sama. Perbedaanya, pada unique key yaitu Anda diizinkan untuk memasukkan nilai NULL.

***CONTOH:***

***CREATETABLE Persons (  
ID int NOTNULLUNIQUE,  
LastName varchar(255) NOTNULL,  
FirstName varchar(255),  
Age int  
);***

47.SQL PRIMARY KEY

Primary Keyadalah sebuah aturan yang dimana fungsinya yaitu untuk membedakan antara baris satu dengan baris lainnya yang ada pada suatu tabel dan mempunyai sifat unik.

***CONTOH:***

***CREATETABLE Persons (***

***ID int NOTNULL,***

***LastName varchar(255) NOTNULL,***

***FirstName varchar(255),***

***Age int,***

***PRIMARYKEY (ID)***

);

48.SQL FOREIGN KEY

Pada suatu batasan Primary Key secara unik yang mengidentifikasi pada setiap record dalam suatu SQL Syntax merupakan Databaseyang paling sering berisi satu atau lebih tabel. Pada setiapakandiidentifikasi dengan nama (misalnya. "penjual" atau "Pesanan"). Tabel berisi record (baris) dengan data.

tabel.Kunci utama harus berisi suatu nilai UNIK, dan tidak boleh berisi suatu nilai NULL.Sebuah tabel hanya dapat memiliki satu kunci utama; dan dalam tabel, kunci utama ini dapat terdiri dari satu atau beberapa kolom (bidang).

***CONTOH:***

***CREATETABLE Orders (***

***OrderID int NOTNULL***

***OrderNumber int NOTNULL***

***PersonID int,***

***PRIMARYKEY (OrderID),***

***FOREIGNKEY (PersonID) REFERENCESPersons(PersonID)  
);***

49.SQL CHECK

Pada suatu Batasan Check yaitu digunakan untuk membatasi suatu kisaran nilai yang dapat kita tempatkan di kolom. Jika kita menentukan batasan Check pada satu kolom, maka itu hanya akan memungkinkan nilai-nilai tertentu untuk kolom tersebut. Jika kita menentukan batasan suatu Check pada tabel, maka itu bisa membatasi suatu nilai pada kolom tertentu berdasarkan nilai di kolom lain pada baris.

***CONTOH:***

***REATETABLE Persons (***

***ID int NOTNULL,***

***LastName varchar(255) NOTNULL,***

***FirstName varchar(255),***

***Age int,***

***CHECK (Age>=18)***  
);

50.SQL DEFAULT

Batasan Default dapat kita gunakan untuk memberikan suatu nilai default untuk kolom. Nilai default akan ditambahkan pada semua rekaman baru apabilah tidak ada nilai lain yang ditentukan.

***CONTOH:***

***CREATETABLE Persons (***

***ID int NOTNULL,***

***LastName varchar(295) NOTNULL,***

***FirstName varchar(295),***

***Age int,***

***City varchar(295) DEFAULT'Sandnes'  
);***

51.SQL INDEX

Auto increment field pada MySQL merupakan suatu tipe field integer yang secara otomatis akan bertambah nilainya jika terjadi penambahan row pada table dimana field tersebut berada.

***CONTOH:***

***CREATETABLE Persons (***

***Personid int NOTNULL AUTO\_INCREMENT,***

***LastName varchar(285) NOTNULL,***

***FirstName varchar(285),***

***Age int,***

***PRIMARYKEY (Personid)  
);***

52.SQL AUTO INCREMENT

Auto increment field pada MySQL merupakan suatu tipe field integer yang secara otomatis akan bertambah nilainya jika terjadi penambahan row pada table dimana field tersebut berada.

***CONTOH:***

***CREATETABLE Persons (***

***Personid int NOTNULL AUTO\_INCREMENT,***

***LastName varchar(285) NOTNULL,***

***FirstName varchar(285),***

***Age int,***

***PRIMARYKEY (Personid)  
);***

53.SQL DATES

Adapun bagian tersulit saat kita bekerja dengan tanggal yaitu memastikan bahwa pad suatu format tanggal yang kita coba sisipkan, cocok dengan format kolom tanggal dalam suatu database

54.SQL VIEWS

View merupakan sebuah perintah query yang disimpan pada suatu database dengan menggunakan nama tertentu, sehingga bisa kita gunakan pada setiap saat untuk melihat data tanpa kita harus menuliskan ulang query tersebut. Kita dapat menambahkan fungsi sql, pernyataan WHERE dan JOIN kedalam suatu view dan menyajikan data dalam AS jika data itu berasal dari satu table.

***CONTOH:***

***CREATEVIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';***

55.SQL INJECTION

InjectionSQLmerupakan suatu teknik injeksi kode yang dapat merusak suatu database kita .Injeksi SQL merupakan salah satu teknik peretasan web yang paling umum.Injeksi SQL juga merupakan penempatan kode yang berbahaya dalam pernyataan SQL, melalui input halaman web.

***CONTOH:***

***uName = getRequestString("username");  
uPass = getRequestString("userpassword");  
sql = 'SELECT \* FROM Users WHERE Name ="' + uName + '" AND Pass ="' + uPass + '"'***

56.SQL HOSTING

Apabilah kita ingin situs web kita dapat menyimpan dan mengambil data-data dari database, maka server web kita harus memiliki akses ke sistem database yang menggunakan bahasa SQL.Jika server web kita dihosting oleh Penyedia Layanan Internet (ISP), kita harus mencari berupa paket hosting SQL.Database SQL hosting yang paling umum adalah MS SQL Server, Oracle, MySQL, dan MS Access

57.SQL TYPES

Pada tipe data kolom akan menentukan nilai apa saja yang dapat dimiliki kolom: karakter, integer,tanggal , uang, dan waktu, biner, dan seterusnya.

58.SQL REFERENCES

SQL (Structured Query Language) merupakan suatu bahasa yang dapatdigunakan untuk menjalankan suatu perintah manipulasi atau dapat juga d

igunakan untuk mengakses data pada database. SQL biasanya disebutnya sebagai Query, dan semua DBMS didunia menggunakan perintah yang dinamakan SQL, istilah inilah yang umum untuk penamaan bahasa database.

59.SQL KEYWORDS

Kata keywords merupakan kata yang dapat kita gunakan untuk melakukan suatu operasi dasa suatu database dan keywords ini juga merupakan kata kunci dalam SQL

60.MY SQL FUNCTIONS

MySQL adalah database engine atau server database yang mendukung pada bahasa database pencarian SQL. MySQL juga merupakan sebuah perangkat lunak system manajemen basis data SQL atau DBMS yang multithread, multi-user. MySQL AB ini membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

**Function** atau kata lainnya method mempunyai fungsi yaitu merupakan suatu sub program yang dapat diperuntukan untuk mengerjakan suatu perintah tertentu yang sesuai pada fungsi method itu sendiri. Didalam **mysql** sudah terdapat beberapa fungsi default yang dapatkita digunakan untuk mengerjakan sutau tugas tertentu.

61.SQL SERVER FUNCTIONS

**Function** atau kata lainnya method mempunyai fungsi yaitu merupakan suatu sub program yang dapat diperuntukan untuk mengerjakan suatu perintah tertentu yang sesuai pada fungsi method itu sendiri.SQL Serverjuga memiliki banyak fungsi bawaan.Referensi ini berisi suatu string, numerik, tanggal, konversi, dan beberapa fungsi lanjutan di SQL Server.

62.MS ACCESS FUNCTIONS

Microsoft Access merupakan sebuah program aplikasi basis data komputer relasional yang ditujukan pada kalangan rumahan dan juga perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, dan juga selain itu tentunya Microsoft Word,Microsoft Excel, dan Microsoft PowerPoint.**Function** atau kata lainnya method mempunyai fungsi yaitu merupakan suatu sub program yang dapat diperuntukan untuk mengerjakan suatu perintah tertentu yang sesuai pada fungsi method itu sendiri.

63.SQL QUICK REF

|  |  |
| --- | --- |
| **SQL Statement** | **Syntax** |
| AND / OR | SELECT column\_name(s) FROM table\_name WHERE condition AND|OR condition |
| ALTER TABLE | ALTER TABLE table\_name ADD column\_name datatype  or  ALTER TABLE table\_name DROP COLUMN column\_name |
| AS (alias) | SELECT column\_name AS column\_alias FROM table\_name  or  SELECT column\_name FROM table\_name  AS table\_alias |
| BETWEEN | SELECT column\_name(s) FROM table\_name WHERE column\_name BETWEEN value1 AND value2 |
| CREATE DATABASE | CREATE DATABASE database\_name |
| CREATE TABLE | CREATE TABLE table\_name ( column\_name1 data\_type, column\_name2 data\_type, column\_name3 data\_type, ... ) |
| CREATE INDEX | CREATE INDEX index\_name ON table\_name (column\_name)  or  CREATE UNIQUE INDEX index\_name ON table\_name (column\_name) |
| CREATE VIEW | CREATE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition |
| DELETE | DELETE FROM table\_name WHERE some\_column=some\_value  or  DELETE FROM table\_name (**Note:**Deletes the entire table!!)  DELETE \* FROM table\_name (**Note:**Deletes the entire table!!) |
| DROP DATABASE | DROP DATABASE database\_name |
| DROP INDEX | DROP INDEX table\_name.index\_name (SQL Server) DROP INDEX index\_name ON table\_name (MS Access) DROP INDEX index\_name (DB2/Oracle) ALTER TABLE table\_name DROP INDEX index\_name (MySQL) |
| DROP TABLE | DROP TABLE table\_name |
| EXISTS | IF EXISTS (SELECT \* FROM table\_name WHERE id = ?) BEGIN --do what needs to be done if exists END ELSE BEGIN --do what needs to be done if not END |
| GROUP BY | SELECT column\_name, aggregate\_function(column\_name) FROM table\_name WHERE column\_name operator value GROUP BY column\_name |
| HAVING | 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 |
| IN | SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1,value2,..) |
| INSERT INTO | INSERT INTO table\_name VALUES (value1, value2, value3,....)  *or*  INSERT INTO table\_name (column1, column2, column3,...) VALUES (value1, value2, value3,....) |
| INNER JOIN | SELECT column\_name(s) FROM table\_name1 INNER JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| LEFT JOIN | SELECT column\_name(s) FROM table\_name1 LEFT JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| RIGHT JOIN | SELECT column\_name(s) FROM table\_name1 RIGHT JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| FULL JOIN | SELECT column\_name(s) FROM table\_name1 FULL JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| LIKE | SELECT column\_name(s) FROM table\_name WHERE column\_name LIKE pattern |
| ORDER BY | SELECT column\_name(s) FROM table\_name ORDER BY column\_name [ASC|DESC] |
| SELECT | SELECT column\_name(s) FROM table\_name |
| SELECT \* | SELECT \* FROM table\_name |
| SELECT DISTINCT | SELECT DISTINCT column\_name(s) FROM table\_name |
| SELECT INTO | SELECT \* INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name  *or*  SELECT column\_name(s) INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name |
| SELECT TOP | SELECT TOP number|percent column\_name(s) FROM table\_name |
| TRUNCATE TABLE | TRUNCATE TABLE table\_name |
| UNION | SELECT column\_name(s) FROM table\_name1 UNION SELECT column\_name(s) FROM table\_name2 |
| UNION ALL | SELECT column\_name(s) FROM table\_name1 UNION ALL SELECT column\_name(s) FROM table\_name2 |
| UPDATE | UPDATE table\_name SET column1=value, column2=value,... WHERE some\_column=some\_value |
| WHERE | SELECT column\_name(s) FROM table\_name WHERE column\_name operator value |