**/8\*SASS:**

1. Biến. $

2. Xếp chồng.

3. Hàm @Mixin($a,$b){}

4.+@include.

5. &hover{}

6. Tính toán.

7. @import “….”

8. @extend Class.

**\*SQL SERVER**

**1. DATABASE :**

**TẠO:** + **CREATE DATABASE** ten\_co\_so\_du\_lieu;

**SHOW**: + **SHOW DATABASE**;

**XÓA**: + **DROP DATABASE** ten\_co\_so\_du\_lieu;

**SỬ DỤNG:** + **USE** ten\_co\_so\_du\_lieu;

**TRUY VẤN:** + **SELECT** danh\_sach\_cot **FROM** ten\_bang

[Menh\_de WHERE]

[Menh\_de GROUP BY]

[Menh\_de HAVING]

[Menh\_de ORDER BY];

**SỬA TÊN:** + **ALTER** **DATABASE** ten\_cu **MODIFY** **NAME** = ten\_moi**;**

**2 . TABLE:**

**TẠO:** + **CREATE** **TABLE** ten\_bang(

kieu\_du\_lieu cot1,

kieu\_du\_lieu cot2,

kieu\_du\_lieu cot3,

.....

kieu\_du\_lieu cotN,

**PRIMARY** **KEY(** mot hoac nhieu cot )

);

**XÓA table**: + **DROP** **TABLE** ten\_bang;

**XÓA hàng**: + **DELETE** **FROM** ten\_bang [**WHERE** dieu\_kien];

+ **DELETE** **FROM** ten\_bang;

**SỬA Tên Cột**: + **SP\_RENAME** 'ten\_cu',’ten\_moi’,’COLUMN’;

**ALTER table**: + **ALTER** **TABLE** ten\_bang **ADD** ten\_cot kieu\_du\_lieu;

+ **ALTER** **TABLE** ten\_bang **DROP** **COLUMN** ten\_cot;

+ **ALTER** **TABLE** ten\_bang **MODIFY** **COLUMN** ten\_cot kieu\_du\_lieu;

+ **ALTER** **TABLE** ten\_bang **ADD** **CONSTRAINT** MyUniqueConstraint UNIQUE(cot1, cot2...);

+ **ALTER** **TABLE** ten\_bang **ADD** **CONSTRAINT** MyUniqueConstraint CHECK (CONDITION);

+ **ALTER** **TABLE** ten\_bang **ADD** **CONSTRAINT** MyPrimaryKey **PRIMARY** **KEY** (cot1, cot2...);

+ **ALTER** **TABLE** ten\_bang **DROP** **CONSTRAINT** MyPrimaryKey;

+ **ALTER** **COLLUM** ten\_cot kieu\_cot;

+ **DROP** **COLLUM** ten\_cot;

**3. THAO TÁC CƠ BẢN:**

**UPDATE**: + Cập nhật lại giá trị của cột 1, cột 2 của các hàng được với ĐIỀU KIỆN

+ **UPDATE** ten\_bang

**SET** cot1 = giatri1, cot2 = giatri2...., cotN = giatriN

**WHERE** [dieu\_kien];

**INSERT**: + **INSERT** **INTO** **TABLE**\_**TEN** (cot1, cot2, cot3,...cotN)]

**VALUES** (giatri1, giatri2, giatri3,...giatriN);

**Có dấu Tiếng Việt** :

**VALUES**( **N’giatri1’** , **N’giatri1’**);

**SELECT:**  + **SELECT** cot1, cot2, cotN **FROM** ten\_bang;

**SELECT** **DISTINCT**:

+ Trong 1 cột có thể có nhiều values giống nhau, câu truy vấn này dùng để lấy các values khác nhau, không bị trùng lại.

+**SELECT DISTINCT** Country **FROM** Customers;

**SELECT INTO:** copy dữ liệu của bản cũ và insert vào 1 table mới

+ **SELECT** cot1, cot2, cotN

**INTO** bang\_moi

**FROM** bang\_cu

**WHERE** dieu\_kien

**ORDER BY**: + **SELECT** danh\_sach\_cot

**FROM** ten\_bang

[**WHERE** dieu\_kien]

[**ORDER** **BY** cot1, cot2, .. cotN] [**ASC** | **DESC**];

**ORDER BY** cot1 **ASC**, cot2 **DESC**;

**GROUP BY**: +Dùng cho các HÀM như **count**(), **sum**(), **avg**(),…….

**4. MỆNH ĐỀ**

**WHERE**: + **WHERE** [dieu kien];(=; >;< ;>= ;<=; <>;(NOT) BETWEEN; LIKE; (not)IN)

+ **AND** || **OR** || **NOT**

**AS**.

**LIKE**: +’%**XXX**% ’

+’X%**’**

+’%X**’**

+’\_X&’

**TOP**: + **SELECT** **TOP** **number**|**percent** ten\_cot

**FROM** ten\_bang

**WHERE** [dieu\_kien]

**IN**: +**SELECT** \* **FROM** Customers

**WHERE** Country **IN | NOT IN** (**SELECT** Country **FROM** Suppliers);

**BETWEEN DATE:** +**WHERE** OrderDate **BETWEEN** '1996-07-01' **AND** '1996-07-31';

**5. HOẠT ĐỘNG SQL NÂNG CAO:**

**NULL**: + **IS** **NOT** **NULL**

+ **IS** **NULL**

+ **NOT** **NULL**

**+ WHERE** Address **IS NOT NULL**

**AS**: + **AS** ten\_alias

**CHỈ MỤC INDEX**: + **CREATE** **INDEX** ten\_chi\_muc **ON** ten\_bang;

+ **CREATE** **INDEX** ten\_chi\_muc **ON** ten\_bang (ten\_cot);

+ **CREATE** **INDEX** ten\_chi\_muc **ON** ten\_bang (cot1, cot2);

+ **DROP** **INDEX** ten\_chi\_muc;

+ LƯU Ý: Khi nào ko nên dung chỉ mục:

\*KHÔNG DÙNG TRONG BẢNG NHỎ.

\*BẢNG THƯỜNG XUYÊN UPDATE, INSERT.

\*KHÔNG NÊN SỬ DỤNG TRÊN CÁC CỘT CÓ NHIỀU GTRI NULL.

\*CÁC CỘT THƯỜNG XUYÊN BỊ SỬA ĐỔI.

**VIEW**: + **CREATE** **VIEW** ten\_view **AS**

**SELECT** cot1, cot2.....

**FROM** ten\_bang

**WHERE** [dieu\_kien];

+ **WITH** **CHECK** **OPTION**;

+ **UPDATE** SINHVIEN\_VIEW

**SET** TUOI = 20

**WHERE** TEN='Hoang';

+ **DELETE** **FROM** SINHVIEN\_VIEW

**WHERE** TUOI = 18;

+ **DROP** **VIEW** SINHVIEN\_VIEW;

**HAVING**: + THỨ TỰ, VỊ TRÍ CHUẨN:

**SELECT** cot1, cot2

**FROM** bang1, bang2

**WHERE** [ cac\_dieu\_kien ]

**GROUP** **BY** cot1, cot2

**HAVING** [ cac\_dieu\_kien ]

**ORDER** **BY** cot1, cot2;

+ Dùng cho các HÀM như **count**(), **sum**(), **avg**(),…….

**TRANSACTION**:

+ **BEGIN** **TRANSACTION**.

+ **SAVEPOINT**: Đánh dấu 1 thời điểm bất kì để **ROLLBACK** trở lại.

**SAVEPOINT** TEN\_CUA\_SAVEPOINT;

**SAVE** **TRANSACTION** TEN\_CUA\_SAVEPOINT;

+ **COMMIT**: Lưu các thay đổi.Đã Commit rồi thì không thể RollBack được nữa.

COMMIT;

+ **ROLLBACK**: quay trở lại trạng thái trước khi thay đổi.

**ROLLBACK** **TRANSACTION** TEN\_CUA\_SAVEPOINT;

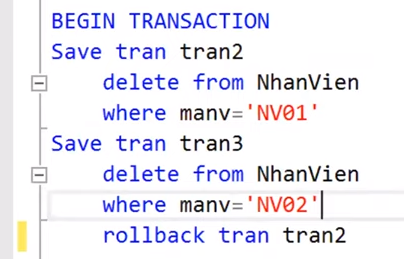
**ROLLBACK** **TO** TEN\_CUA\_SAVEPOINT;

**ROLLBACK**;

+ **SET** **TRANSACTION**: đặt một tên cho TRANSACTION.

**SET** **TRANSACTION** [ READ WRITE | READ ONLY ];

+ **RELEASE** **SAVEPOINT** TEN\_CUA\_SAVEPOINT;



**WILDCARD**: + %: Một hoặc nhiều ký tự.

+ \_: Một ký tự.

**Truy vấn**

**lông nhau**: + **SELECT**

**SELECT** ten\_cot [, ten\_cot ]

**FROM** bang1 [, bang2 ]

**WHERE** ten\_cot TOAN\_TU

(**SELECT** ten\_cot [, ten\_cot ]

**FROM** bang1 [, bang2 ]

[**WHERE**])

+ **INSERT (Insert vào 1 bản khác )**

**INTO** ten\_bang [ (cot1 [, cot2 ]) ]

**SELECT** [ \*|cot1 [, cot2 ]

**FROM** bang1 [, bang2 ]

[ **WHERE** GIA\_TRI TOAN\_TU ]

+ **UPDATE**

**UPDATE** table

**SET** ten\_cot = gia\_tri\_moi

[ WHERE TOAN\_TU [ GIA\_TRI ]

(**SELECT** TEN\_COT

**FROM** TEN\_BANG)

[ **WHERE**) ]

+ **DELETE**

**DELETE** FROM TEN\_BANG

[ WHERE TOAN\_TU [ VALUE ]

(SELECT TEN\_COT

FROM TEN\_BANG)

[ WHERE) ]

**SEQUENCE**: + **IDENTITY**(1,1)

+ Thay đổi cột Identity mới:

**DBCC** **CHECKIDENT(‘**table**\_**name'**,** RESEED**,** new**\_**reseed**\_value);**

**6. RÀNG BUỘc SQL**

**NOT NULL**: + Bảo đảm một cột không thể có giá trị NULL.

**DEFAULT**: + Cung cấp một giá trị mặc định cho cột khi không được xác định.

+ HOCPHI DECIMAL (18, 2) DEFAULT 3500000,

**UNIQUE**: + Bảo đảm tất cả giá trị trong một cột là khác nhau.

**PRIMARY KEY**: + Mỗi hàng/bản ghi được nhận diện một cách duy nhất trong một bảng

+ **ALTER TABLE tên\_bảng**

**ADD CONSTRAINT PK\_tên PRIMARY KEY (ID, TEN);**

**Or: ALTER TABLE tên\_bảng**

**ADD PRIMARY KEY (ID);**

**+ ALTER TABLE tên\_bảng  CONSTRAINT DROP tên\_PK ;**

**FOREIGN KEY**: + Mỗi hàng/bản ghi được nhận diện một cách duy nhất trong bất kỳ

bảng nào.

+ **CONSTRAINT** **tên\_FK** **FOREIGN KEY** **(tên\_cột)**  
     **REFERENCES** **Persons(tên\_cột\_PK)**

**+ ALTER TABLE Orders  
 ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);**

+ **ALTER TABLE tên\_bảng  
 DROP CONSTRAINT tên\_FK;**

**CHECK**: + Bảo đảm tất cả giá trị trong một cột thỏa mãn các điều kiện nào đó.

+ id int **CHECK (điều kiện);**

+ **ALTER TABLE tên\_bảng  
 ADD CHECK (điều kiện);**

**+ ALTER TABLE tên\_bảng  
 DROP CONSTRAINT tên\_cột\_có\_check;**

**INDEX**: + Sử dụng để tạo và lấy dữ liệu từ Database một cách nhanh chóng.

**+ CREATE INDEX *index\_name*  
 ON *table\_name* (*column1*, *column2*, ...);**

**+ DROP INDEX *table\_name*.*index\_name*;**

**XÓA RÀNG BUỘC**: **+ ALTER TABLE EMPLOYEES DROP CONSTRAINT EMPLOYEES\_PK;**

**7. JOIN**

**JOIN**: + Kết hợp các cột lại với nhau từ hai bảng bằng cách sử dụng giá trị chung của mỗi bảng.

**INNER JOIN**: + Lấy cái chung.

+ **FROM** bảng\_1 **INNER JOIN** bảng \_2

**ON** id\_bảng\_1 = id\_bảng\_2

**LEFT JOIN**: + Điền các cột bảng **PHẢI vào bảng TRÁI.**

**+ FROM** bang1 **LEFT JOIN** bang2

**ON** bang1.field\_chung = bang2.field\_chung;

**RIGHT JOIN**: + Điền các cột bảng **TRÁI vào bảng PHẢI.**

**+ FROM** bang1 **RIGHT JOIN** bang2

**ON** bang1.field\_chung = bang2.field\_chung;

**FULL JOIN**: + Lấy cả hai bảng

+  **FROM** bang1 FULL **JOIN** bang2

**ON** bang1.field\_chung = bang2.field\_chung;

**8. HÀM TRONG SQL:**

**Hàm hữu ích**: + Hàm **COUNT(ten\_cot);**

+ Hàm **MAX**(ten\_cot);

+ Hàm **MIN**(ten\_cot);

+ Hàm **AVG**(ten\_cot);

+ Hàm **SUM**(ten\_cot);

+ Hàm **RAND**();

+ Hàm **CONCAT**();

**Hàm DATE**: + https://vietjack.com/sql/ham\_xu\_ly\_date\_trong\_sql.jsp

**Hàm xử lý chuỗi**:+ <https://vietjack.com/sql/ham_xu_ly_chuoi_trong_sql.jsp>

**Hàm xử lý số**: + https://vietjack.com/sql/ham\_xu\_ly\_so\_trong\_sql.jsp