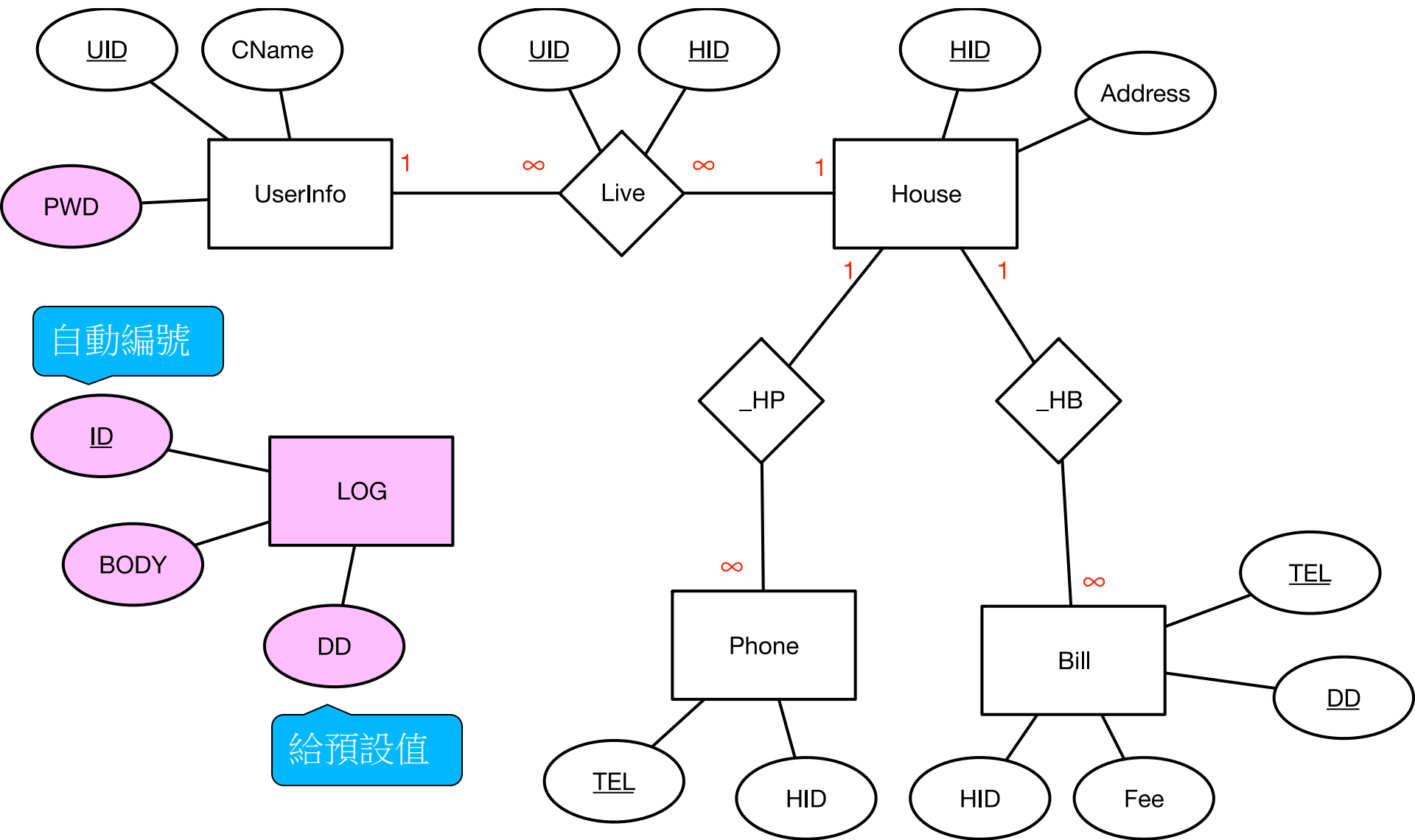


# Trigger

---

朱克剛



# Trigger（觸發）

---

功用:攔截資料表中發生的INSERT、DELETE與UPDATE事件  
語法:

```
CREATE TRIGGER tr_trigger_name  
ON table_name  
FOR INSERT, DELETE, UPDATE  
AS  
BEGIN  
    // trigger 觸發後要做的事情寫這裡  
END
```



BEGIN  
END  
可省略

# 範例

若 UserInfo 中新增一筆資料，將此事件自動記錄到 Log 資料表中

```
CREATE TRIGGER tr_log_userinfo
ON UserInfo
FOR INSERT
AS
    DECLARE @uid NVARCHAR(50)
    DECLARE @cname NVARCHAR(50)
```

```
-- 停止計算 SQL 影響的資料筆數
SET NOCOUNT ON
```

```
SELECT @uid = uid, @cname = cname FROM INSERTED
INSERT INTO Log (body) VALUES
    ('在資料表 UserInfo 中新增 ' + @uid + ', ' + @cname + ' 資料')
```

存放新增或修改的那一整筆資料

注意cname為null

# 包山包海的 Trigger

---

## 區別INSERT、DELETE、UPDATE

```
CREATE TRIGGER tr_log_userinfo  
ON UserInfo  
FOR INSERT, DELETE, UPDATE  
AS
```

```
IF exists(SELECT * FROM deleted) and exists(SELECT * FROM inserted)  
    print('修改資料')
```

```
ELSE IF exists(SELECT * FROM deleted) and not exists(SELECT * FROM inserted)  
    print('刪除資料')
```

```
ELSE IF not exists(SELECT * FROM deleted) and exists(SELECT * FROM inserted)  
    print('新增資料')
```

```
ELSE  
    print('無資料變動')
```

# UPDATE()

---

若新密碼與帳號一樣則讓密碼更新失敗

```
CREATE TRIGGER tr_userinfo_pwd ON UserInfo FOR UPDATE
AS
  DECLARE @uid NVARCHAR(50)
  DECLARE @pwd NVARCHAR(50)
  IF update(pwd)
  BEGIN
    SELECT @uid = uid, @pwd = pwd FROM inserted
    IF @pwd = @uid
    BEGIN
      print ('密碼不可與帳號一樣')
      ROLLBACK
    END
  END
END
```

# @@ROWCOUNT

---

此為系統變數，儲存最近一筆異動資料的異動筆數  
不可使用 `set nocount on` 指令，否則會得到 0  
若執行 `delete from UserInfo`

```
select @@rowcount
```

```
||
```

```
select @n = count(*) from deleted  
select @n
```

# Instead of Trigger

---

用來取代 Insert、Update 與 Delete 指令的 trigger

```
CREATE TRIGGER tr_userinfo_delte ON UserInfo
INSTEAD OF DELETE
AS
BEGIN
    Select 'Someone executes delete command' as Message
END
```



# 不會觸發遞迴

---

Instead of trigger 不會觸發遞迴效應

例如，使用 trigger 擋住一次修改兩筆以上資料

```
create trigger tr_userinfo_delete
on UserInfo instead of delete
as
begin
    if @@rowcount = 1
    begin
        delete from temp where uid in (select uid from deleted)
    end
end
```

# 遞迴 (recursive)

---

## 直接遞迴

修改 TableA 資料 → trigger → 修改 TableA 資料

## 間接遞迴

修改 TableA 資料 → trigger1 → 修改 TableB 資料

修改 TableB 資料 → trigger2 → 修改 TableA 資料

避免遞迴情況發生

# 查詢 Trigger

---

列出資料庫中所有的 Trigger

```
select  
    OBJECT_NAME(parent_id) as Table_Name,  
    *  
from AddressBook.sys.triggers
```

# 停用或啟用 Trigger

---

停用 disable



Trigger name

disable trigger all on userinfo

啟用 enable

enable trigger all on userinfo

# 重複觸發程序

---

遞迴層級最大可達 32 層

透過「重複觸發程序」選項開啟「直接遞迴」功能

- 該資料庫 → 滑鼠右鍵 → 屬性 → 選項

該選項只能關閉「直接遞迴」

在遞迴中只要執行到 **rollback** 指令，遞迴到此結束