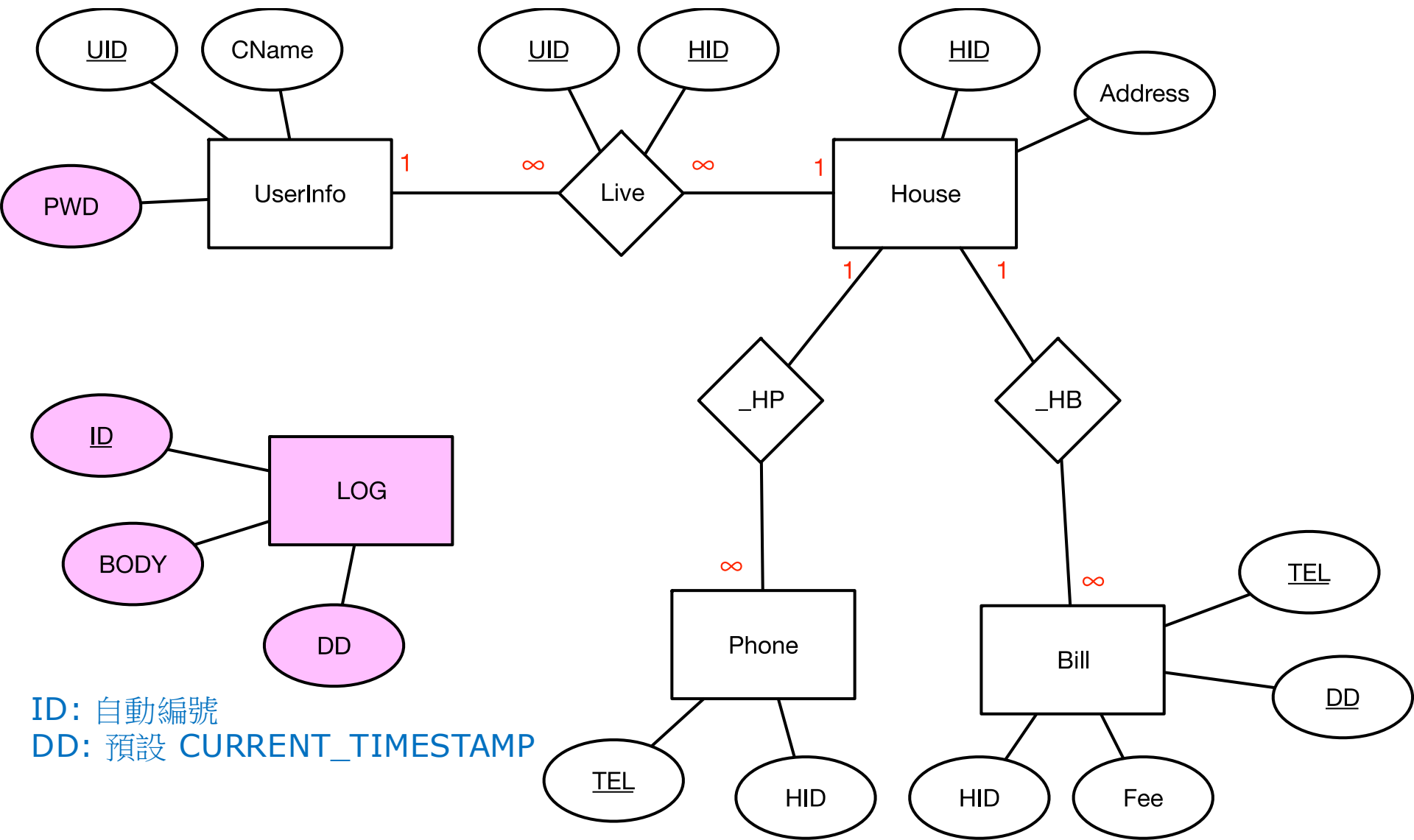


Trigger

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:

```
DELIMITER $$
```

```
CREATE TRIGGER tr_trigger_name
```

```
AFTER INSERT
```

```
ON table_name FOR EACH ROW
```

```
BEGIN
```

```
-- trigger 觸發後要做的事情寫這裡
```

```
END $$
```

```
DELIMITER ;
```

BEFORE / AFTER

INSERT
DELETE
UPDATE

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

delimiter \$\$

create trigger **tr_log_userinfo_insert** after insert on UserInfo for each row
begin

```
    set @body = concat('將 [, new.uid, ', ', new.cname, '] 加到userinfo資料表中');  
    insert into Log (body) values (@body);  
end $$
```

delimiter ;

```
set @a = null;  
select concat('hello', convert(ifnull(@a, 'abc'), char));
```

NEW 與 OLD

暫存表 NEW 表示新增的資料

暫存表 OLD 表示刪除的資料

想一想，修改資料時 NEW 與 OLD 分別儲存什麼資料？

中斷 trigger

範例：擋住一次兩筆以上的資料異動

若更動兩筆，此 trigger 會觸發兩次，此時 @count 相當於整體變數，每次累加，因此第二次就會發出 signal

```
CREATE trigger tr_userinfo_update before update
on UserInfo for each row
BEGIN
    set @count = if (@count IS NULL, 1, (@count+1));
    if @count > 1 then
        signal sqlstate '45001' set message_text = 'Stop!!';
    end if;
END;
```

Spaceship 運算子

如何得知修改哪一個欄位，例如修改的 password 欄位

old.password <=> **new**.password

傳回 true 表示 password 沒有修改

傳回 false 表示有修改

查詢與刪除 trigger

查詢

SHOW TRIGGERS

刪除

DROP TRIGGER tr_trigger_name

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

練習

請將 UserInfo 資料表中的任何資料異動都 Log 起來，並在 Log 中詳細記錄異動資料內容