


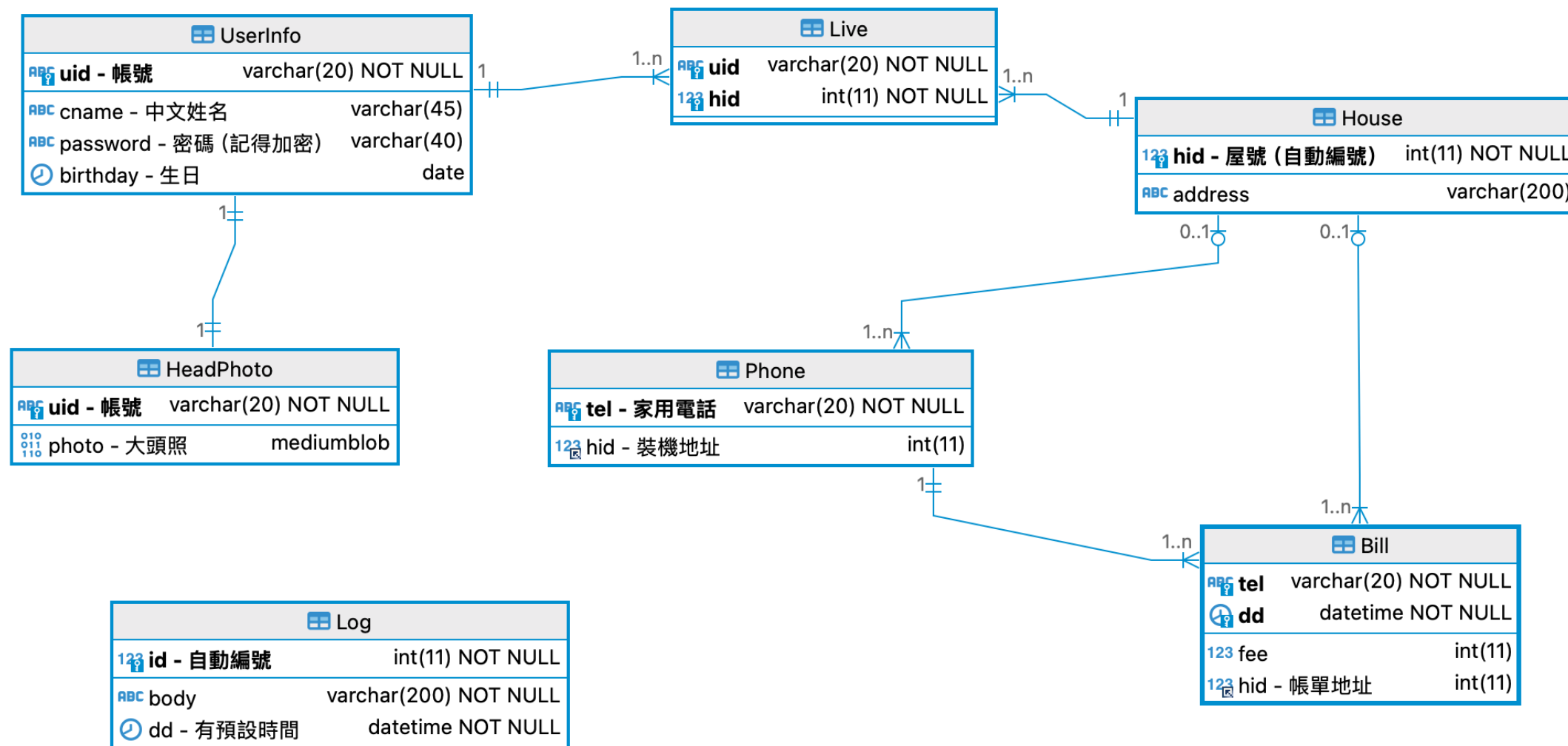


# SQL COMMAND

朱克剛



# ERD + Data Dictionary



# SELECT基本語法

SELECT 欄位

FROM 資料表

WHERE 條件

沒有條件時，  
WHERE 可不寫

# SELECT 單一資料表

- 顯示 UserInfo 資料表中的所有資料

```
SELECT *  
FROM UserInfo
```

- 顯示 UserInfo 資料表中的個別欄位

```
SELECT uid, cname  
FROM UserInfo
```

# LIKE

- 需要所有姓李的基本資料

```
SELECT *  
FROM UserInfo  
WHERE cname LIKE '李%'
```

SQL command 的字串  
為單引號

- 列出所有姓李的以及身份證字號後四碼為1234

```
SELECT *  
FROM UserInfo  
WHERE cname LIKE '李%' AND uid LIKE '%1234'
```

=, >, <, >=, <=, <>

- 列出電話費超過500元的資料

```
SELECT *  
FROM Bill  
WHERE fee > 500
```

- 列出電話費在500~1000元的資料

```
SELECT *  
FROM Bill  
WHERE fee >= 500 AND fee < 1000
```

# BETWEEN AND

- 列出電話費 500 - 1000 元的資料

```
SELECT *  
FROM Bill  
WHERE fee BETWEEN 500 AND 1000
```

包含 500 與 1000

# IN 、 NOT IN

- 列出王大明與李大媽的基本資料

```
SELECT *  
FROM UserInfo  
WHERE cname IN ('王大明', '李大媽')
```

||

```
SELECT *  
FROM UserInfo  
WHERE cname = '王大明' OR cname = '李大媽'
```



# 排序 ORDER BY

## ■ 將電話號碼由小到大排序

```
SELECT *  
FROM Phone  
ORDER BY tel
```

## ■ 將電話號碼由大到小排序

```
SELECT *  
FROM Phone  
ORDER BY tel DESC
```

# MYSQL按照中文筆畫數排序

- MySQL預設編碼為utf8，若要按照中文筆畫數排序，轉成big5再排即可

```
SELECT *  
FROM UserInfo  
ORDER BY convert(cname using big5)
```

# IS NULL 、 IS NOT NULL

## ■ 列出欠缺姓名資料的使用者

```
SELECT *  
FROM UserInfo  
WHERE cname IS NULL OR cname = ''
```

## ■ 不要這樣寫（速度超慢）

```
SELECT *  
FROM UserInfo  
WHERE ifnull(cname, '') = ''
```

MySQL: ifnull()  
SQL Server: isnull()

# 不喜歡看到 NULL 流出

## ■ SQL Server

```
SELECT uid, isnull(cname, "")  
FROM UserInfo
```

## ■ MySQL

```
SELECT uid, ifnull(cname, "")  
FROM UserInfo
```

# 函數 count()

- 列出UserInfo資料表中有多少筆資料

```
SELECT count(*)  
FROM UserInfo
```

# 關連線處理 - JOIN

## ■ 四種類型

- *INNER JOIN*

- 關連線左右兩邊均有相同的資料

- *LEFT OUTER JOIN*

- 關連線左側資料種類較多

- *RIGHT OUTER JOIN*

- 關連線右側資料種類較多

- *CROSS JOIN*

- 未設定關連而形成交叉合併

## ■ *FULL JOIN = LEFT + RIGHT* 但不是每家資料庫都有

# INNER JOIN

- 列出身份證字號、姓名、住址、電話

```
SELECT
    UserInfo.uid, cname, address, tel
FROM
    UserInfo, Live, House, Phone
WHERE
    UserInfo.uid = Live.uid AND
    Live.hid = House.hid AND
    House.hid = Phone.hid
```

# OUTER JOIN

- 列出身份證字號、姓名、住址、電話

```
SELECT
    Userinfo.uid, cname, address, tel
FROM
    Userinfo LEFT JOIN Live
        ON UserInfo.uid = Live.uid
    LEFT JOIN House
        ON Live.hid = House.hid
    LEFT JOIN Phone
        ON House.hid = Phone.hid
```



# 群組 GROUP BY

- 列出每支電話的總費用

```
SELECT tel, sum(fee)
FROM Bill
GROUP BY tel
```

# 別名

## ■ 列出身份證字號、姓名、住址、電話

```
SELECT
    a.uid AS '身份證字號',
    cname AS '姓名',
    address AS '住址',
    tel AS '電話'
FROM
    UserInfo AS a, Live AS b, House AS c, Phone AS d
WHERE
    a.uid = b.uid AND
    b.hid = c.hid AND
    c.hid = d.hid
ORDER BY a.uid
```

# 不重複資料 DISTINCT

## ■ 列出所有的姓氏

```
SELECT DISTINCT left(cname, 1)
FROM UserInfo
```

補充 left(), right(), mid()

## ■ 列出每個姓氏有幾筆資料

```
SELECT lastname, count(*) AS n
FROM (
    SELECT left(cname, 1) as lastname
    FROM UserInfo
) AS a
GROUP BY lastname
```

# TOP ( SQL Server )

- 請列出繳費金額最高的前三支電話號碼

```
SELECT TOP 3 *  
FROM Bill  
ORDER BY fee DESC
```

- 請列出前百分之50的會員資料

```
SELECT TOP 50 PERCENT *  
FROM UserInfo
```

# LIMIT (MySQL)

- 請列出繳費金額最高的前三支電話號碼

```
SELECT * FROM Bill ORDER BY fee DESC  
LIMIT 3
```

- SKIP 功能

```
SELECT * FROM UserInfo  
LIMIT 2, 3
```

跳過 2 筆抓三筆

# 極端值查詢

## ■ 列出電話費最高的帳單資料

```
select *  
from Bill  
where fee = (select max(fee) from Bill)
```

## ■ 列出帳單金額總和最高的那支電話

```
select X.*  
from (  
    select tel, sum(fee) as n from Bill group by tel  
) as X, (  
    select max(n) as n  
    from (  
        select tel, sum(fee) as n from Bill group by tel  
    ) as tmp  
) as Y  
where X.n = Y.n
```

# 練習

- 列出每個人有多少支電話？
- 列出每個人有多少地方住？
- 列出那間屋子住最多人？
- 計算空屋率。

# UNION ALL

- 垂直合併兩個結構一樣的查詢結果

```
SELECT * FROM UserInfo WHERE uid = 'A01'  
UNION ALL  
SELECT 'B01', 'David'
```



# CASE WHEN

- 帳單金額如果超過 500 就打 8 折，超過 300 打 9 折

```
SELECT fee,  
  case  
    when fee > 500 then fee * 0.8  
    when fee > 300 then fee * 0.9  
    else fee  
  end  
FROM Bill
```

# HAVING

- 列出平均繳費金額超過300元的電話資料

```
SELECT tel, avg(fee)
FROM Bill
GROUP BY tel
HAVING avg(fee) > 300
```



```
SELECT * FROM (
    SELECT tel, avg(fee) AS avg_fee
    FROM Bill
    GROUP BY tel
) AS a
WHERE avg_fee > 300
```

# View

- 建立一個住在台北市民眾資料的視觀表

```
CREATE VIEW vw_livein_taipei AS
  SELECT UserInfo.uid, cname, address
  FROM UserInfo, Live, House
  WHERE UserInfo.uid = Live.uid AND Live.hid = House.hid
  AND address LIKE '台北市%'
```

- 使用

```
SELECT * FROM vw_livein_taipei
```

# View 的優點

- View 的優點一大堆：
- 少了語法檢查時間，執行速度快
- 屏蔽資料庫 Schema
- 可控制資料揭露程度
- 建立時若沒下特殊指令，View 為唯讀
- 後端工程師會很開心 ❤️

# INSERT INTO

- 插入一筆新的資料到 UserInfo 資料表

```
INSERT INTO UserInfo VALUES (  
    'B01', 'David', NULL, NULL, NULL  
)
```

```
INSERT INTO UserInfo (uid, cname) VALUES (  
    'B02', 'Tom'  
)
```

# UPDATE

- 更新資料表中所有資料 ( 別亂下!!! )

```
UPDATE UserInfo SET cname = NULL
```

- 更新特定資料

- 將A03 的姓名改為孫小毛，身份證字號改為Z01

```
UPDATE UserInfo SET cname = '孫小毛', uid = 'Z01'  
WHERE uid = 'A03'
```

# DELETE

- 刪除所有電話帳單資料 ( 別亂下!!! )

DELETE FROM **Bill**

或

TRUNCATE TABLE **Bill**

- 刪除孫小毛資料

DELETE FROM **UserInfo** WHERE uid = 'Z01'