

### การเชื่อมโยงแปลงที่ดินกับฐานข้อมูล

อีกหนึ่งความสามารถของโปรแกรม POWER CIVIL : DTM ก็คือการเชื่อมโยงข้อมูลระหว่าง AutoCAD Drawing กับฐานข้อมูล ซึ่งในที่นี้ เราใช้โปรแกรม Microsoft Office Access เพราะเป็นโปรแกรมที่ใช้งานได้ง่าย และส่วนมาก จะมาพร้อมกับการติดตั้งโปรแกรม Microsoft Office อื่น ๆ ซึ่งในที่นี้ เราจะสามารถนำเข้าข้อมูลที่อยู่ในรูปแบบตารางเข้าสู่ Drawing และนำข้อมูลส่งออกจาก Drawing กลับไปยังฐานข้อมูลได้ และในกรณีที่ข้อมูลแปลงที่ดินมีจำนวนมาก ก็สามารถใช้คุณสมบัติการค้นหาโดยระบุเงื่อนไขได้

### ฐานข้อมูล (Database) คืออะไร?

ฐานข้อมูล คือการออกแบบจัดเก็บข้อมูลอย่างเป็นระเบียบเพื่อง่ายต่อการจัดการและนำไปใช้งานอื่น ๆ ซึ่งข้อมูลจะถูกจัดเก็บอยู่ในรูปแบบของตาราง และคอลัมน์ต่าง ๆ ซึ่งแบ่งตามประเภทของข้อมูล เช่น ตารางข้อมูลแปลงที่ดิน ก็จะประกอบด้วยคอลัมน์ เลขที่แปลงที่ดิน, ชื่อนามสกุลผู้ครอบครองกรรมสิทธิ์, ตำแหน่งที่ตั้ง, โทรศัพท์, พื้นที่, การใช้งาน, สถานะการชำระภาษี เป็นต้น เมื่อเราจัดเก็บข้อมูลอยู่ในรูปแบบของฐานข้อมูล จะทำให้เราสามารถใช้งานคุณสมบัติ SQL Command ซึ่งเป็นภาษาที่ใช้ในการจัดการฐานข้อมูล ทั้งการค้นหา เพิ่มแถว แก้ไข ลบข้อมูลได้

### SQL Command คำสั่งที่ใช้ในการจัดการฐานข้อมูล

เมื่อข้อมูลที่ถูกจัดเก็บในตารางมีจำนวนมากขึ้นการจัดการกับข้อมูลก็ทำได้ยากขึ้น ดังนั้นเราจึงใช้คำสั่ง SQL ในการจัดการกับฐานข้อมูลต่าง ๆ

### พื้นฐาน คำสั่ง SELECT

ในระบบฐานข้อมูลนั้น ข้อมูลทั้งหมดจะถูกจัดเก็บไว้ในรูปของ Table และ Columns ตัวอย่างเช่น ใน Table แปลงที่ดิน เราจัดเก็บข้อมูล LandID, Name, Surname, Area, Building, TaxAmount, Paid

## ตัวอย่าง Database

LandIdInfor						
LandId	Name	Surname	Area	Building	Tax Amount	Paid
1	ชนะพันธ์	อินทรเกษตร	250	ไม้	500	No
2	สมชาย	เป็นสุข	150	บ้าน	30	Yes
3	สมชาติ	สนอง	150	บ้าน	10	No
4	ชนะพันธ์	อิน	150	ไม้	100	No
5	วิทยา	เจริญ	317	บ้าน	20	Yes
7	กมล	ทำดี	200	โรงงาน	20	No
8	ประกอบ	รักดี	100	ที่เปล่า	30	No

จากข้อมูลในตาราง ถ้าเราต้องการแสดงชื่อนามสกุลทุกคนโดยใช้คำสั่ง "Select" SQL Command จะเป็นดังนี้

```
SELECT Name, Surname FROM LandIdInfor
```

จาก SQL Command ข้างบนเราจะได้ผลดังนี้

Name	Surname
ชนะพันธ์	อินทรเกษตร
สมชาย	เป็นสุข
สมชาติ	สนอง
ชนะพันธ์	อิน
วิทยา	เจริญ
กมล	ทำดี
ประกอบ	รักดี

จากคำสั่งด้านบน เราได้ร้องขอข้อมูลทั้งหมดใน LandIdInfor โดยที่เราระบุ Columns ที่เราต้องการไปด้วย ใน Command ดังนั้น ข้อมูลที่เราจะได้อีกก็จะถูกจัดเรียงตาม Columns ที่เราส่งไปกับคำสั่ง SELECT นั่นเอง คำสั่ง Select แบบทั่วไป

```
SELECT ColumnName1,ColumnName2, ...  
FROM TableName;
```

นอกจากนี้ในคำสั่ง SQL เราสามารถที่จะขอข้อมูลทั้งหมดทุก Columns ใน Table โดยใช้เครื่องหมาย \* แทน การพิมพ์ชื่อทุก Columns ในกรณีนี้ข้อมูลก็จะถูกส่งมาโดยเรียงตามลำดับ Columns ที่ระบุไว้ใน โครงสร้าง ของ Tables

```
SELECT * FROM TableName;
```

ในการใช้ คำสั่ง SELECT\* นั้น เราควรใช้ในกรณีที่เราแน่ใจจริงๆว่าเราต้องการข้อมูลทั้งหมดใน ทุก Columns มิฉะนั้นแล้วจะทำให้ Database ทำงานเกินความจำเป็น อีกทั้งเวลาในการประมวลผลคำสั่งนาน นั้นยังมากกว่า การใช้ SELECT Command โดยระบุ Column name อีกด้วย อย่างไรก็ตาม ใน Database แต่ละตัวก็อาจจะมีโครงสร้างของคำสั่งที่แตกต่างไปบ้างเล็กน้อย แต่โดยทั่วไปแล้ว พื้นฐานคำสั่งหลัก ๆ ก็เหมือนๆ กัน

### คำสั่ง SELECT แบบมีเงื่อนไข

ก่อนที่เราจะพูดถึงรายละเอียด มาดูตัวอย่าง Table และเครื่องหมายในการเปรียบเทียบก่อน

EmployeeTable			
EmployeeID	Salary	Benefits	Position
010	75000	15000	Manager
105	65000	15000	Manager
152	60000	15000	Manager
215	60000	12500	Manager
244	50000	12000	Staff
300	45000	10000	Staff
335	40000	10000	Staff
400	32000	7500	Entry- Level
441	28000	7500	Entry- Level

## เครื่องหมายเปรียบเทียบใน SQL

โดยทั่วไปแล้ว เครื่องหมายเปรียบเทียบ ที่ใช้ใน Database หลัก ๆ จะมีอยู่ด้วยกัน 6 ชนิด คือ

=	เท่ากับ
<> หรือ !=	ไม่เท่ากับ
<	น้อยกว่า
>	มากกว่า
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ

ในภาษา SQL นั้นเราสามารถสร้างเงื่อนไขของข้อมูลที่เราต้องการได้ โดยใช้เครื่องหมายเปรียบเทียบที่เราต้องการร่วมกับ คำสั่ง WHERE โดยใส่ไว้ข้างหลังคำสั่ง SELECT เสมอ โดยเราจะเรียกในส่วนเงื่อนไขนี้ว่า Where clause ดังตัวอย่าง

จาก Table ตัวอย่างด้านบน สมมติว่าเราต้องการ ดูข้อมูล EmployeeID ของคนที่มี salary มากกว่า 50,000 SQL Command ก็จะเป็นดังนี้

```
SELECT EmployeeID
FROM EmployeeTable
WHERE SALARY >= 50000;
```

โดยเครื่องหมาย >= (มากกว่า หรือ เท่ากับ) จะดึงข้อมูลของ Records ที่มีข้อมูลในcolumn salary มากกว่า หรือ เท่ากับ 50000 เท่านั้น มาแสดงผล ดังนั้น ผลที่จะแสดงจะเป็นดังนี้:

```
EmployeeID
-----
010
105
152
215
244
```

นอกจากนี้เรายังสามารถใช้ คำสั่ง Where clause กับ columns ที่เก็บข้อมูลเป็นข้อความได้อีกด้วย ตัวอย่าง เราต้องการ แสดงผลข้อมูลของคนที่มีตำแหน่ง "Manager" SQL Command ก็จะเป็นดังนี้

```
SELECT EmployeeID
FROM EmployeeTable
WHERE POSITION = 'Manager';
```

อย่างไรก็ตาม ในการใช้เครื่องหมาย เปรียบเทียบกับ Column ที่เป็นข้อความนั้น โดยส่วนมากจะใช้ได้กับ เครื่องหมาย มากกว่า, เท่ากับ หรือ คำสั่ง Like เท่านั้น โดยค่าของตัวแปรที่เราส่งเข้าไป จะต้องมีการใช้เครื่องหมาย Single Quote (') หรือ Double Quote (") ก่อนและหลังตัวแปรเสมอ

การใช้เครื่องหมายเปรียบเทียบมากกว่า 1 ตัว

คำสั่ง AND เป็นคำสั่งในการเชื่อมโยงเงื่อนไขที่เราต้องการข้อมูลที่มีเงื่อนไขมากกว่า 1 อย่าง โดย ข้อมูลที่จะแสดงผลนั้น จะต้องตรงกับเงื่อนไขทั้งหมด ที่เราส่งไปที่ Database ตัวอย่างเช่น, เราต้องการ ID ของ คนที่มีรายได้มากกว่า 40,000 และมี Position เท่ากับ "Staff": SQL Command จะเป็นดังนี้

```
SELECT EMPLOYEEID
FROM EMPLOYEEETABLE
WHERE SALARY > 40000 AND POSITION = 'Staff';
```

คำสั่ง OR เป็นคำสั่งที่ใช้ในการเชื่อมโยงเงื่อนไขเหมือนกับ คำสั่ง AND แต่จะต่างกันตรงที่ ข้อมูลที่จะได้นั้น สามารถ เป็นข้อมูลที่ตรงกับเงื่อนไขใดเงื่อนไขหนึ่งก็ได้ ตัวอย่างเช่น, เราต้องการ ID ของคนที่ได้ Salary น้อยกว่า 40,000 หรือ มี benefits น้อยกว่า 10,000 in benefits SQL Command จะเป็นดังนี้

```
SELECT EMPLOYEEID
FROM EMPLOYEEETABLE
WHERE SALARY < 40000 OR BENEFITS < 10000;
```

นอกจากนี้ คำสั่ง AND และคำสั่ง OR ยังสามารถใช้ด้วยกันได้อีกด้วย ตัวอย่างเช่น

```
SELECT EMPLOYEEID
FROM EMPLOYEEETABLE
WHERE POSITION = 'Manager' AND SALARY > 60000 OR BENEFIT > 12000;
```

จากชุดคำสั่งข้างบน อย่างแรก SQL จะค้นหาข้อมูลของคนที่มีตำแหน่ง ='Manager'จากนั้นก็ใช้ผลลัพธ์นั้นหาต่อไปว่ามี records ใดบ้างที่มี ข้อมูลใน column SALARY มากกว่า60000. แล้วจากนั้นจึง ไปหาข้อมูลในตารางใหม่ว่ามี Record ใดบ้างที่มี BENEFITS มากกว่า12000 บ้าง จากนั้นจึงเอาผลลัพธ์ทั้งสองมารวมกับ. ดังนั้นข้อมูลทั้งหมดที่เราจะได้จาก คำสั่งข้างบนจะเป็นข้อมูล ของคนที่มีตำแหน่ง เป็น Manager และ ได้เงินเดือนมากกว่า 60000นอกจากนี้ยังรวมไปถึงข้อมูลของคนอื่นที่ไม่ได้เป็น Manager แต่ได้ BENEFITS มากกว่า12000 อีกด้วย

อย่างไรก็ตาม ในกรณีที่เรต้องการสร้างเงื่อนไขมากกว่า 1 เงื่อนไขเราควรที่จะใส่ เครื่องหมาย ( และ ) ก่อนและหลังเงื่อนไข นั้น เพื่อ ง่าย แก่การเข้าใจ และ ตัวอย่าง เช่น .

```
SELECT EMPLOYEEID
FROM EMPLOYEE
WHERE (POSITION = 'Manager' AND SALARY > 60000) OR BENEFIT > 12000;
```

### คำสั่ง BETWEEN

คำสั่ง BETWEEN เป็นคำสั่งที่ใช้ในการสร้างเงื่อนไขที่มากกว่า 1 เงื่อนไข ในคำสั่งเดียว ตัวอย่างเช่น เราต้องการ ข้อมูลของคนที่มีเงินเดือนระหว่าง 30,000 - 50,000 เราสามารถเขียน ได้ดังนี้

```
SELECT EMPLOYEEID
FROM EMPLOYEE
WHERE SALARY BETWEEN 30000 AND 50000
```

นอกจากนี้เรายังสามารถใช้คำสั่ง NOT ร่วมกับ IN หรือ BETWEEN เพื่อที่จะได้ผลตรงข้ามกัน ดังเช่น

```
SELECT EMPLOYEEID
FROM EMPLOYEE
WHERE SALARY NOT BETWEEN 30000 AND 50000
```

### คำสั่ง LIKE

เป็นการค้นหาข้อมูลใน Column ที่เป็นข้อความ ซึ่งถ้าแปลตรงตัว คำว่า Like จะมีความหมายว่าเหมือนหรือคล้าย ดังนั้น เมื่อนำมาใช้ใน SQL Command จะใช้ในการค้นหาเงื่อนไขที่คล้ายกับค่าที่กำหนด ตัวอย่างเช่น ถ้าเราต้องการค้นหาชื่อที่นำหน้าด้วย 'ส' จากตารางด้านล่าง จะต้องเขียน SQL Command ดังนี้

```
SELECT * FROM LandIdInfor
```

WHERE Name Like 'ส\*'

LandIdInfor						
LandId	Name	Surname	Area	Building	Tax Amount	Paid
1	ธนะ พันธ์	อินทรเกษตร	250	ไม้	500	No
2	สมชาย	เป็นสุข	150	บ้าน	30	Yes
3	สมชาติ	สนอง	150	บ้าน	10	No
4	ธนะ พันธ์	อิน	150	ไม้	100	No
5	วิทยา	เจริญ	317	บ้าน	20	Yes
7	กมล	ทำดี	200	โรงงาน	20	No
8	ประกอบ	รักดี	100	ที่เปล่า	30	No

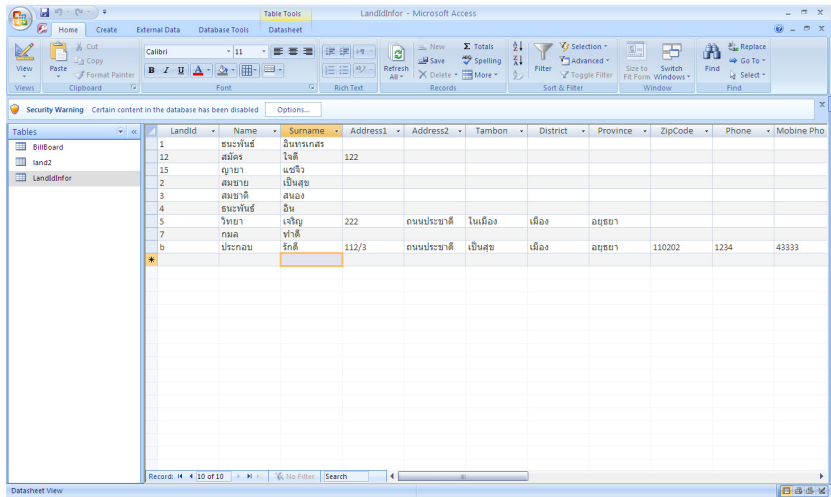
ผลการค้นหาจะออกมาดังนี้

LandId	Name	Surname	Area	Building	Tax Amount	Paid
2	สมชาย	เป็นสุข	150	บ้าน	30	Yes
3	สมชาติ	สนอง	150	บ้าน	10	No

สังเกตว่า องค์กรประกอบของคำสั่งในเงื่อนไข จะเป็น WHERE ColumnName Like '\*ค่าที่ต้องการ\*' ซึ่งเครื่องหมายดอกจัน (\*) สามารถวางได้ทั้งด้านหน้าและหลังข้อความ ถ้ามีด้านหลังอย่างเดียว เช่น 'ส\*' หมายถึง ต้องนำหน้าด้วย ส โดยไม่สนใจว่าอักษรที่ตามหลังจะเป็นตัวอะไร ถ้ามีทั้งหน้าและหลัง '\*ส\*' ก็หมายถึง ประกอบด้วย ส ในคอลัมน์นั้น ๆ

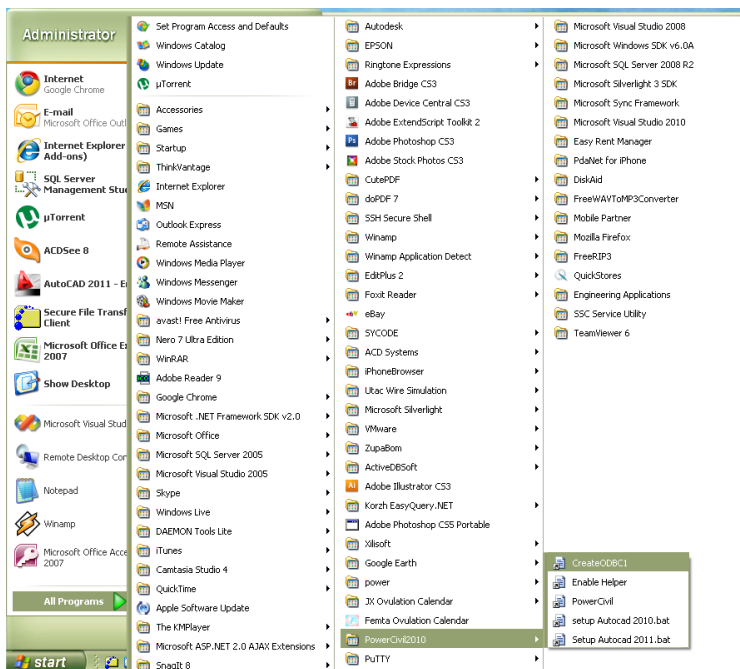
### ฐานข้อมูลที่ใช้งานในโปรแกรม

ในตัวอย่างที่มากับโปรแกรม จะใช้ Database Microsoft Access ซึ่งจะอยู่ใน C:\Powercivil\data\db.mdb และเมื่อเปิดมา จะมีตารางที่เราใช้งานคือ LandIdInfor ซึ่งในส่วนนี้ เราสามารถนำเข้าข้อมูลที่มีอยู่แล้ว อาจจะอยู่ในรูปแบบของ Excel มาใส่ในตารางนี้ ก่อนที่จะใช้งาน



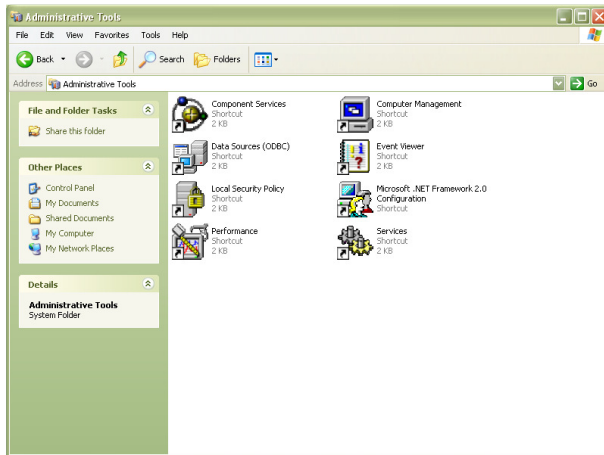
## เริ่มการทำงานครั้งแรก

เนื่องจากในการติดต่อระหว่างโปรแกรม Powercivil และฐานข้อมูล Microsoft Access จำเป็นต้องมีตัวกลางในการติดต่อก็คือ ODBC ซึ่งเป็น Driver ที่ใช้ในการควบคุมเกี่ยวกับฐานข้อมูลต่าง ๆ ซึ่งโปรแกรมได้มีการสร้างให้อัตโนมัติ ใน Start -> Program -> PowerCivil -> CreateODBC1 ดังรูป

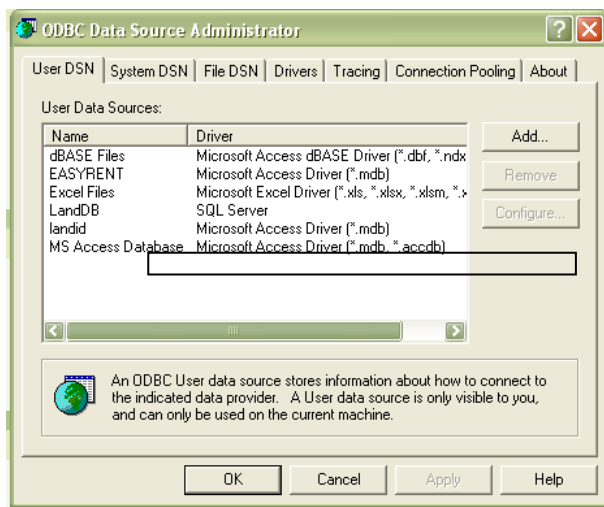


เมื่อกดแล้ว ให้ไปที่ Control Panel -> Administrative Tools -> Data Sources (ODBC) หรือ Start -> Run แล้วพิมพ์ `odbcad32` แล้ว Enter จะเจอกับหน้าต่าง

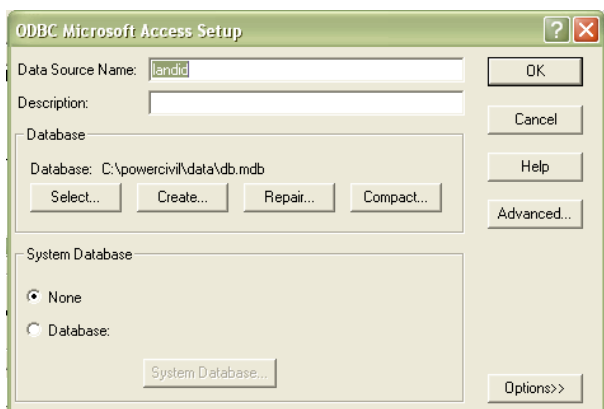




เมื่อเราดับเบิลคลิกเปิด ODBC จะเจอกับหน้าต่างดังนี้




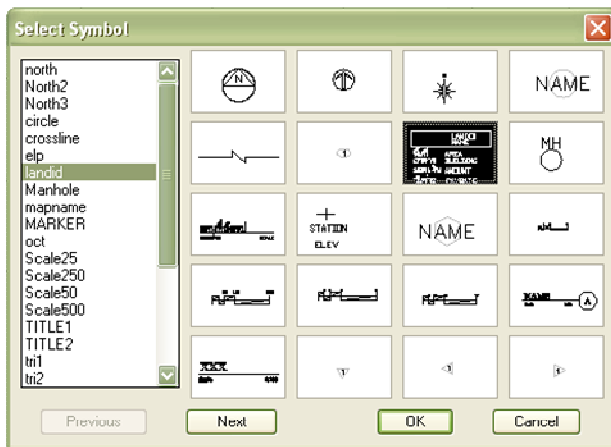
จะเห็นว่า เมื่อสักครู่นี้เราดคำสั่ง CreateODBC1 นั้น ได้ทำการสร้าง landid ขึ้นให้เราเรียบร้อยแล้ว ถ้าต้องการดูด้านใน ให้เลือกที่ landid แล้วเลือก Configure..




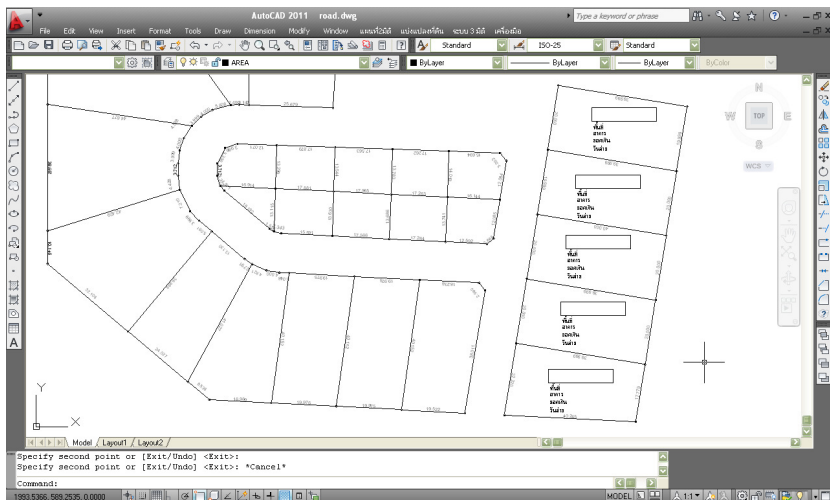
ที่ Database จะชี้ไปยังตำแหน่งของตัวอย่างที่มากับโปรแกรม คือ C:\powercivil\data\db.mdb ซึ่งถ้าเราทำการสร้าง Database ตัวใหม่ ให้มาทำการแก้ไขค่าที่นี้ โดยเลือกที่ Select แล้วเลือกชี้ไปยัง Database ที่เราต้องการ

เมื่อทำการตั้งค่าต่าง ๆ เรียบร้อยแล้ว ให้กลับมาที่โปรแกรม Powercivil ตัวอย่างเราจะใช้แปลงที่ดินนำเข้า และแบ่งแปลงไว้ก่อนหน้า

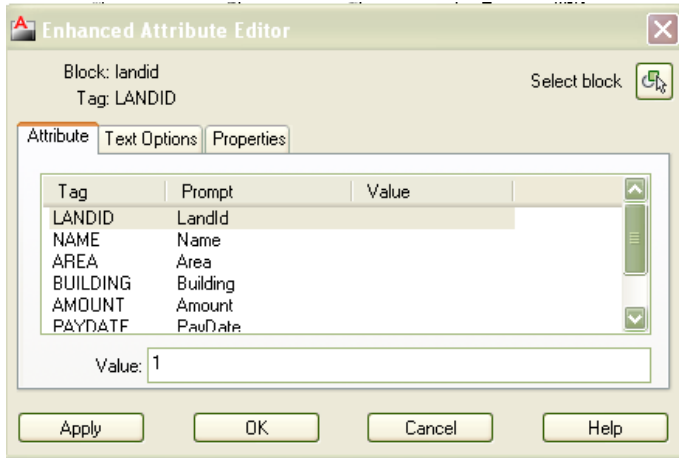
สัญลักษณ์ที่เราจะใช้ในการแสดงข้อมูลแปลงที่ดิน เราจะเลือกไปที่ไอคอน  (สัญลักษณ์แผนที่) หรือจากแถบเมนู แผนที่ 2 มิติ -> สัญลักษณ์แผนที่ จะแสดงหน้าต่าง “Select Symbol




จะขึ้นหน้าต่างให้เลือกสัญลักษณ์ที่ต้องการ ในที่นี้เราเลือก landid เสร็จแล้วนำมาวางในแปลงที่ดินที่ได้แบ่งไว้ วางบล็อก landid ในแปลงที่ต้องการ แล้วใช้เครื่องมือ Copy  วางตามแปลงต่าง ๆ ดังรูปตัวอย่าง

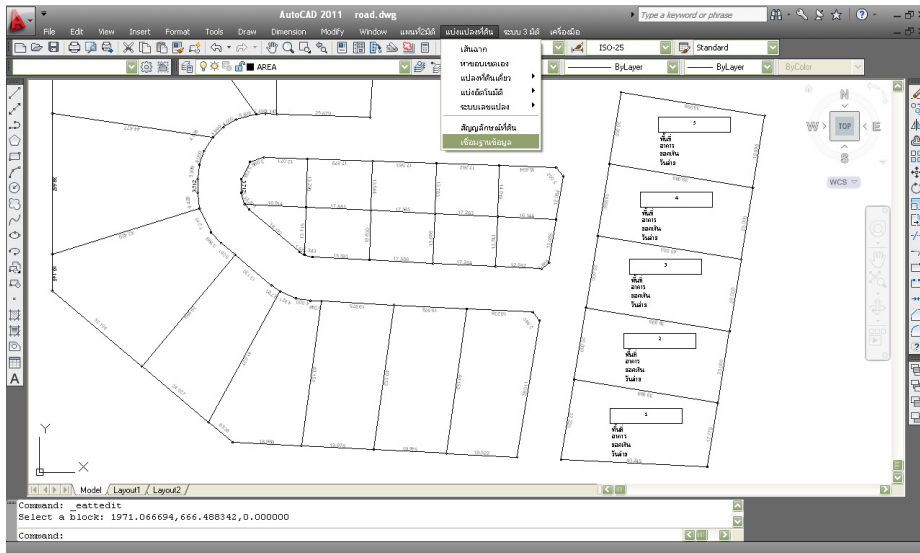


เมื่อวางแล้ว ทำการดับเบิลคลิกที่บล็อก เพื่อแก้ไขเลขที่ของแปลงนั้น ๆ ให้ตรงกับในฐานข้อมูล โดยคลิกเลือกที่ LandId แล้วเปลี่ยนค่าในช่อง Value แล้วกด OK แก้ไขให้ครบทุกบล็อก ซึ่งในตัวอย่างเราจะใช้เลขที่ 1, 2, 3, 4 และ 5



## เชื่อมฐานข้อมูล

เมื่อแก้ไขเสร็จแล้ว จะเป็นขั้นตอนในการนำเข้าสู่ข้อมูลจากฐานข้อมูลให้เลือกที่ไอคอน  (เชื่อมฐานข้อมูล) หรือจากแถบเมนู แบ่งแปลงที่ดิน -> เชื่อมฐานข้อมูล




จะปรากฏหน้าต่าง “เชื่อมโยงข้อมูลภาษี” ประกอบด้วยสองส่วนคือการนำเข้าส่งออกฐานข้อมูล และส่วนของการค้นหา

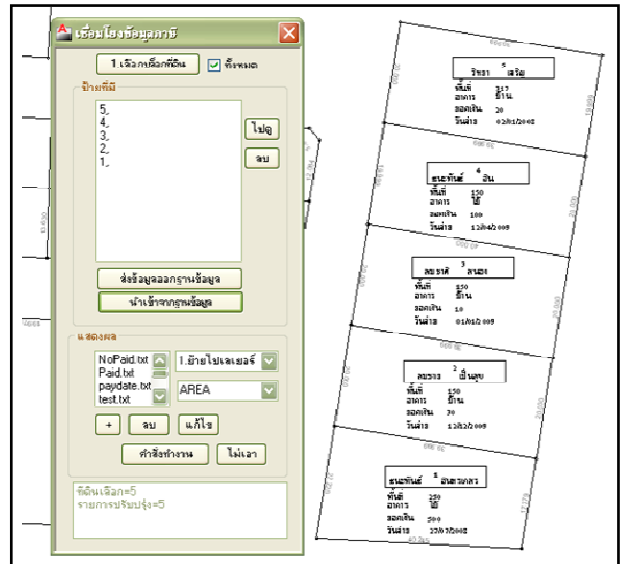
ขั้นตอนแรกเราต้องนำข้อมูลเข้าสู่ Drawing ก่อน ให้กดที่ 1.เลือกบล็อกที่ดิน โดยดูว่า ในช่องทั้งหมดทางด้านขวาถูกเลือกไว้ หมายถึงว่า เราเลือกบล็อก landid ทั้งหมดที่อยู่ใน Drawing เมื่อเรากดแล้ว เลขที่แปลงที่ดินที่เราใส่ไว้จะปรากฏในช่อง ให้กดที่ นำเข้าจากฐานข้อมูล แล้วปิดหน้าต่างนี้ ไปดูที่ บล็อก จะปรากฏข้อมูลตรงตามฐานข้อมูล โดยเชื่อมโยงจาก landid ที่เราได้กำหนดไว้



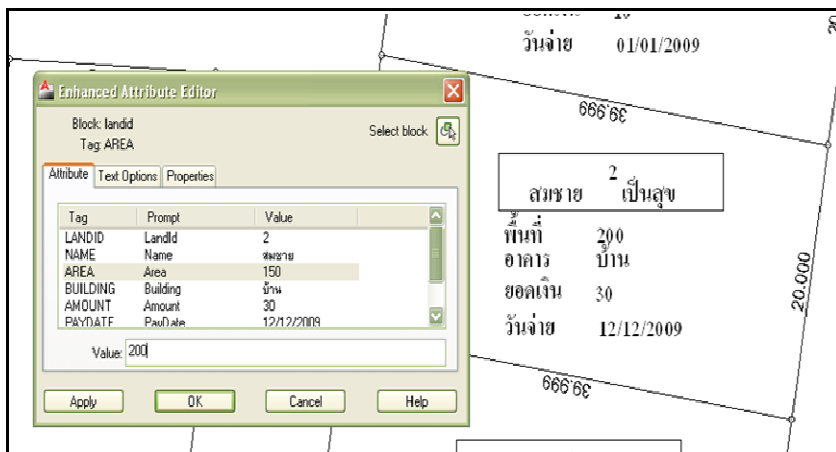
สังเกตว่า ข้อมูลจะเข้าสู่บล็อกตามเลขที่ของ landid ที่เราได้กำหนดไว้ในตอนแรก

ต่อมา นอกจากการนำเข้าแล้ว เรายังสามารถส่งออกข้อมูลที่แก้ไขใน Drawing กลับไปที่ Database โดยวิธีแก้ไขก็ให้ทำการดับเบิลคลิกบล็อกที่ต้องการ แก้ไขค่า และกลับไป หน้าต่าง

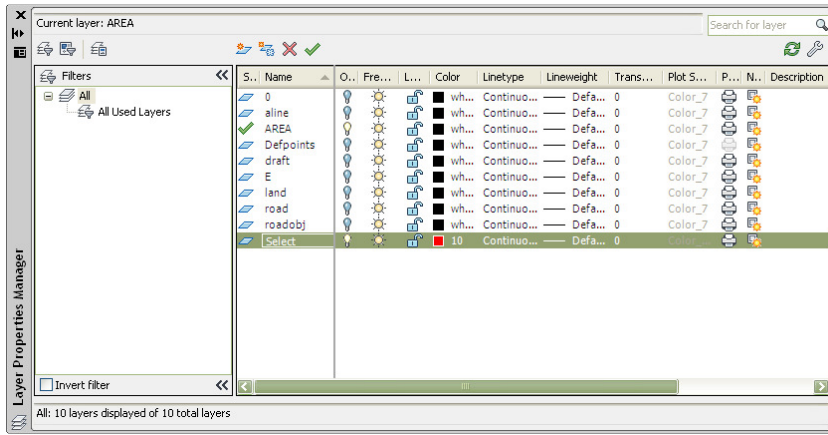
“เชื่อมโยงข้อมูลภาษี” ที่ไอคอน  (เชื่อมฐานข้อมูล) ในครั้งนี้เราต้องการแก้ไขเฉพาะบล็อกที่ เราต้องการเท่านั้น ดังนั้นให้คลิกเอาเครื่องหมายถูกที่ ทั้งหมดออก แล้ว กดเลือกบล็อกที่ดิน เมาส์จะ



กลายเป็นเครื่องหมายให้เลือก ให้ทำการเลือกบล็อกที่แก้ไข แล้วกด Enter เลขที่ของบล็อกที่เลือกก็จะปรากฏในช่อง ให้เรา กด ส่งข้อมูลออกฐานข้อมูล ข้อมูลในฐานข้อมูลก็จะถูกแก้ไข



จะเห็นว่า ในตัวอย่าง เรามีแปลงที่ดินเพียง 5 แปลง ซึ่งไม่ใช่เรื่องยากที่เราจะเลื่อนไปมาเพื่อดูแปลงที่ดินที่ต้องการ แต่ในการใช้งานจริงนั้น เรามีแปลงที่ดินจำนวนมาก ซึ่งการค้นหาและจัดการจะทำได้ยาก ดังนั้นเราจึงมีคุณสมบัติในการค้นหา โดยใช้ SQL Command ที่ได้อธิบายไปในตอนต้น ในการค้นหาข้อมูลที่ต้องการ และใช้หลักการในการเปลี่ยนเลเยอร์ เพื่อให้เห็นผลการค้นหาอย่างชัดเจน เช่นตัวอย่างที่ได้ยกมา จะเป็นการค้นหาแปลงที่ดินที่ยังไม่ได้ชำระภาษี เมื่อค้นหาแล้ว เราจะย้ายไปยังเลเยอร์ที่สร้างใหม่ไว้ เป็นสีแดง ก่อนที่จะเริ่ม ให้ไปทำการสร้างเลเยอร์ใหม่ก่อน ที่ Layer Manager กด New Layer ขึ้นมา ตั้งชื่อว่า Select แล้วเปลี่ยนเป็นสีแดง

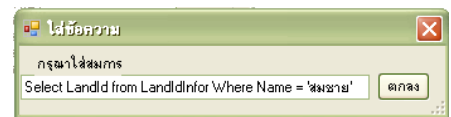


จากนั้น ไปที่ไอคอน (เชื่อมฐานข้อมูล) อีกครั้งกดเลือกบล็อกที่ดินทั้งหมด ในส่วนการค้นหาด้านล่าง ให้เลือก NoPaid.txt เลือก ย้ายไปเลเยอร์และด้านล่าง เลือก Select ซึ่งก็คือเลเยอร์ที่เราสร้างไว้เมื่อสักครู่ จากนั้นกดคำสั่งทำงาน

เมื่อกลับไปดูที่ Drawing จะเห็นว่า บล็อกที่ยังไม่ได้ชำระภาษีจะกลายเป็นสีแดง ซึ่งมาจากในฐานข้อมูล คอลัมน์ Paid ที่เท่ากับ No นั่นเอง นอกจากนี้เงื่อนไขการค้นหาที่มีมาให้ เรายังสามารถเพิ่มเงื่อนไขได้เอง โดยใช้คำสั่ง SQL ก่อนอื่นให้ทำการย้ายบล็อกที่อยู่ทีเลเยอร์ Select กลับไปยัง Layer เดิมก่อน (ในตัวอย่างคือเลเยอร์ AREA)



จากนั้นให้เรากลับไปไปที่ไอคอน (เชื่อมฐานข้อมูล) กดเลือกบล็อกที่ดินทั้งหมด และในส่วนการแสดงผลด้านล่าง ให้เรากดเครื่องหมาย (+) จะปรากฏช่องให้เรากรอกชื่อคำสั่งให้เราใส่ว่า "สมชาย" จากนั้นกดตกลง จะปรากฏช่องให้ใส่คำสั่ง SQL ในกรณีนี้ เราจะทำการสร้างเงื่อนไขที่ค้นหาเฉพาะคนที่ชื่อสมชาย ดังนั้น คำสั่งที่เราใช้จึงเป็น **Select LandId from LandIdInfor Where Name = 'สมชาย'** จากนั้นกดตกลง คำสั่งในช่องจะเพิ่มคำว่า สมชาย.txt มาให้ ให้เราเลือก สมชาย.txt เลือกย้ายไปเลเยอร์ และเลือก Select และกดคำสั่งทำงาน เมื่อกลับไปดูที่ Drawing เราจะเห็นว่า บล็อกที่ดินแปลงนายสมชาย จะเปลี่ยนเป็นสีแดง



นอกจากคำสั่งที่ได้ยกตัวอย่างไปแล้ว ยังสามารถประยุกต์ใช้ได้อีกหลากหลาย เช่นใช้ค้นหาคนที่มีเนื้อที่มากกว่า 100 ไร่ เป็นต้น ซึ่ง คำสั่ง SQL ที่ใช้ในการค้นหา สามารถดูได้จากตอนต้น ซึ่งคุณสมบัติการเชื่อมโยงฐานข้อมูลนี้ ไม่ได้ใช้ได้กับเพียงแค่แปลงที่ดินเท่านั้น แต่ยังสามารถประยุกต์ใช้ได้กับงานด้านอื่น ๆ อีกมากมาย