

Introduction to QuickSQL

- Oracle Apex 提供 QuickSQL 工具, 協助開發者快速建立資料庫表格。
- QuickSQL 提供類似 Markdown 的簡寫語法(Markdown-like shorthand syntax)來產生 DDL 語句。

Example

1. Teacher entity

- id
- name
- email

2. QuickSQL code

```
TEACHER
  id int
  teach_name
  teach_email
```

3. 產生的 DDL 語句

```
create table teacher (
  id
  teach_name
  teach_email
);

number not null constraint teacher_id_pk primary key,
varchar2(255),
varchar2(255)
```

1. 建立表格與欄位

表格稱與欄位

G1. 表格名稱寫在新的一行。

G2. 縮排屬於表格的欄位名稱

Example

```
TEACHER  
  name  
  email
```

主鍵

- 系統會自動產生主鍵 (Primary Key), 名稱為 `id`.
 - 包含 Primary Key 限制條件 (constraint)
- Insert row 時，系統會自動產生主鍵值。

產生的 DDL 如下:

```
-- create tables
create table TEACHER (
    id          number generated by default on null as identity
               constraint TEACHER_id_pk primary key,
    name        varchar2(255 char),
    email       varchar2(255 char)
);
```

手動指定主鍵

若要自己設定主鍵欄位名稱, 使用欄位指令(Column Directive) `/pk` 標記該欄位。

Example

```
TEACHER
  tea_id int /pk [手動指定主鍵]
  name
  email
```

The screenshot displays the QuickSQL interface with a table definition on the left and the generated SQL code on the right.

Table Definition (Left Panel):

```
1 TEACHER
2   tea_id int /pk
3   name
4   email
```

Generated SQL (Right Panel):

```
1 -- create tables
2
3 create table TEACHER (
4     tea_id number generated by default on null as identity
5     |      | constraint TEACHER_tea_id_pk primary key,
6     name varchar2(255 char),
7     email varchar2(255 char)
8 );
9
10
11
```

欄位資料型態

- 若只有寫名稱, 系統會自動設定為 `VARCHAR2(255 char)`
- 若要指定不定長度字串, 或是其他資料型態, 可以使用簡寫語法。

資料型態列表

簡寫語法	資料型態
num, number	NUMBER
int, integer	INTEGER
d, date	DATE
ts, timestamp	TIMESTAMP
char, vc, varchar, varchar2, string	VARCHAR2(4000)
vcNNN	VARCHAR2(NNN) NNN 介於 1 到 32767 之間
vc(NNN)	VARCHAR2(NNN) NNN 介於 1 到 32767 之間
vc32k	VARCHAR2(32767)

資料型態列表 (續)

簡寫語法	資料型態
clob	CLOB
blob	BLOB
file	新增 BLOB 欄位和 _FILENAME、_CHARSET、_MIMETYPE、_LASTUPD 欄位，以便透過瀏覽器上傳或下載檔案。

Example: 開立一個 media 表格, 以儲存上傳的檔案。

MEDIA
image file

↑ Utilities \ Quick SQL

Quick SQL Help Settings ⋮

SQL Diagram

```
1 MEDIA
2 image file [圖片]
```

```
1 -- create tables
2
3 create table MEDIA (
4     id                number generated by default on null as identity
5     |                |                |                |                | constraint MEDIA_id_pk primary key,
6     image             blob,
7     image_filename    varchar2(255 char),
8     image_mimetype    varchar2(255 char),
9     image_charset     varchar2(255 char),
10    image_lastupd     date
11 );
12
13 comment on column MEDIA.image is '圖片';
14
15
16 -- Generated by Quick SQL undefined 10/6/2024, 8:33:17 PM
17
```

2. 欄位限制條件

限制條件類型及層級

限制條件包括:

- 主鍵 (Primary Key)
- 唯一鍵 (Unique Key)
- 非空 (Not Null)
- 檢查 (Check): 限制欄位值必須符合特定條件
- 外鍵 (Foreign Key)

限制條件的套用層級:

- 欄位(單一欄位): 只套用在該欄位
- 表格(多個欄位組合): 套用在多個欄位, 例如: 兩個欄位組成的複合主鍵

單一欄位限制條件

例如, 在客戶表格中:

- 欄位 `email` 必須是唯一值。
- 名稱 `name` 不可為空。
- 區域 `region` 必須是 台北, 台中, 台南, 高雄 之一。
- 信用 `credit` 必須介於 0 到 100 之間。

QuickSQL 語法如下:

CUSTOMER

email vc255 /unique

name /nn [客戶名稱]

region vc20 /check 台北, 台中, 台南, 高雄 [地區]

credit /between 0 and 100 [信用評等]

↑ Utilities \ Quick SQL

Quick SQL

Help Settings ⋮

```
1 CUSTOMER
2   email vc255 /unique
3   name /nn [客戶名稱]
4   region vc20 /check 台北, 台中, 台南, 高雄 [地區]
5   credit /between 0 and 100
```

SQL Diagram

```
1 -- create tables
2
3 create table CUSTOMER (
4     id          number generated by default on null as identity
5     |           | constraint CUSTOMER_id_pk primary key,
6     email       varchar2(255 char)
7     |           | constraint CUSTOMER_email_unq unique,
8     name        varchar2(255 char) not null,
9     region      varchar2(20 char) constraint CUSTOMER_region_ck
10    |           | check (region in ('台北','台中','台南','高雄')),
11    credit       varchar2(4000 char) constraint CUSTOMER_credit_bet
12    |           | check (credit between 0 and 100)
13 );
14
15 comment on column CUSTOMER.name is '客戶名稱';
16 comment on column CUSTOMER.region is '地區';
17
```

常用的限制條件快捷語法 (欄位指令)

快捷語法	說明
/unique	唯一鍵
/nn, /not null	非空
/check	檢查是否在限定的清單值中, 例如: /check 台北, 台中, 台南, 高雄
/between	檢查是否在指定範圍內, 例如: /between 1, 100

上述皆屬欄位指令(Column Directive), 必須寫在欄位名稱後面。

欄位註解

- 使用方括號或是雙破折號來註解欄位。
- 註解寫在限制條件後面。
- 系統會自動產生 `COMMENT ON COLUMN` 語句。

3. 表格層級限制條件

複合唯一鍵

例如, 在上述的客戶表格中, 要求 name 及 company_name 組成複合唯一鍵。

```
CUSTOMER /unique name, company_name  
email vc255 /unique  
name /nn [客戶名稱]  
company_name vc20  
region vc20 /check 台北, 台中, 台南, 高雄 [地區]  
credit /between 0 and 100
```

`/unique` 寫在表格名稱後面, 並指定欄位名稱, 用逗號分隔。

放在表格名稱後的指令稱為表格指令(Table Directive)。

The screenshot displays the Quick SQL interface with two panels: 'Quick SQL' on the left and 'SQL' on the right. The 'Quick SQL' panel shows a table definition for 'CUSTOMER' with columns 'email', 'name', 'company_name', 'region', and 'credit'. The 'name' column is highlighted with a red box, and the text `/unique name, company_name` is shown next to it. The 'SQL' panel shows the corresponding SQL code. The `alter table CUSTOMER add constraint CUSTOMER_uk unique (NAME, COMPANY_NAME);` line is highlighted with a red box.

```
Quick SQL
```

```
1 CUSTOMER /unique name, company_name
2   email vc255 /unique
3   name /nn [客戶名稱]
4   company_name vc20
5   region vc20 /check 台北, 台中, 台南, 高雄 [地區]
6   credit /between 0 and 100
7
```

```
SQL
```

```
1 -- create tables
2
3 create table CUSTOMER (
4     id          number generated by default on null as identity
5     |          |          |          | constraint CUSTOMER_id_pk primary key,
6     email       varchar2(255 char)
7     |          |          |          | constraint CUSTOMER_email_unq unique,
8     name        varchar2(255 char) not null,
9     company_name varchar2(20 char),
10    region      varchar2(20 char) constraint CUSTOMER_region_ck
11    |          |          |          | check (region in ('台北','台中','台南','高雄')),
12    credit       varchar2(4000 char) constraint CUSTOMER_credit_bet
13    |          |          |          | check (credit between 0 and 100)
14 );
15
16 alter table CUSTOMER add constraint CUSTOMER_uk unique (NAME, COMPANY_NAME);
17
18 comment on column CUSTOMER.name is '客戶名稱';
19 comment on column CUSTOMER.region is '地區';
20
```

註: QuickSQL 無法產生複合主鍵, 需要手動修改產生的 DDL 語句。

表格註解

- 在表格指令後面寫註解, 系統會自動產生 `COMMENT ON TABLE` 語句。
- 使用 `--` 或是方括號來註解。

The screenshot displays the Quick SQL interface with two panels. The left panel, titled 'Quick SQL', contains a table creation command for a table named 'CUSTOMER'. The command includes constraints for unique names, email, and region, and a credit limit. A red box highlights the comment '[表格註解]' at the end of the first line. The right panel, titled 'SQL', shows the corresponding SQL code. It includes a comment '-- create tables' and the 'create table' statement with various constraints. A red box highlights the comment 'comment on table CUSTOMER is '表格註解';' at the end of the SQL code.

```
Quick SQL
```

```
1 CUSTOMER /unique name, company_name [表格註解]
2   email vc255 /unique
3   name /nn [客戶名稱]
4   company_name vc20
5   region vc20 /check 台北, 台中, 台南, 高雄 [地區]
6   credit /between 0 and 100
7
```

```
SQL
```

```
1 -- create tables
2
3 create table CUSTOMER (
4     id                number generated by default on null as identity
5     |                 | constraint CUSTOMER_id_pk primary key,
6     email             varchar2(255 char)
7     |                 | constraint CUSTOMER_email_unq unique,
8     name              varchar2(255 char) not null,
9     company_name      varchar2(20 char),
10    region             varchar2(20 char) constraint CUSTOMER_region_ck
11    |                 | check (region in ('台北','台中','台南','高雄')),
12    credit             varchar2(4000 char) constraint CUSTOMER_credit_bet
13    |                 | check (credit between 0 and 100)
14 );
15
16 alter table CUSTOMER add constraint CUSTOMER_uk unique (NAME, COMPANY_NAME);
17
18 comment on table CUSTOMER is '表格註解';
19 comment on column CUSTOMER.name is '客戶名稱';
20 comment on column CUSTOMER.region is '地區';
21
```

4. 外鍵 (Foreign Key)

外鍵的概念

- 外鍵是一個欄位, 參考另一個表格的主鍵。
- 例如, 一個老師可以教多門課, 而一門課只能由一位老師教。
 - 老師與課程之間的關係是一對多的關係。
 - 在多方(課程)的表格中, 使用外鍵參考一方(老師)的主鍵。
 - 一的那方稱為父表格(parent table), 多的那方稱為子表格(child table)。
 - 因為多的那方的外鍵參考了一的那方的主鍵。

程)。

- 系統會自動在子表格中建立外鍵參考父表格的主鍵。

```
Teacher
  name
  email
  Course
    name
    credit
```

↑ Utilities \ Quick SQL

Quick SQL Help Settings

SQL Diagram

```
1 Teacher
2   name
3   email
4   Course
5     name
6     credit
7
```

```
1 -- create tables
2
3 create table Teacher (
4   id          number generated by default on null as identity
5   |          | constraint Teacher_id_pk primary key,
6   name        varchar2(255 char),
7   email       varchar2(255 char)
8 );
9
10
11 create table Course (
12   id          number generated by default on null as identity
13   |          | constraint Course_id_pk primary key,
14   Teacher_id  number
15   |          | constraint Course_Teacher_id_fk
16   |          | references Teacher,
17   name        varchar2(255 char),
18   credit      varchar2(4000 char)
```

Hung Yi-Chen, Dept. of Info. Mgt., CYUT | 2024

建立多對多關係

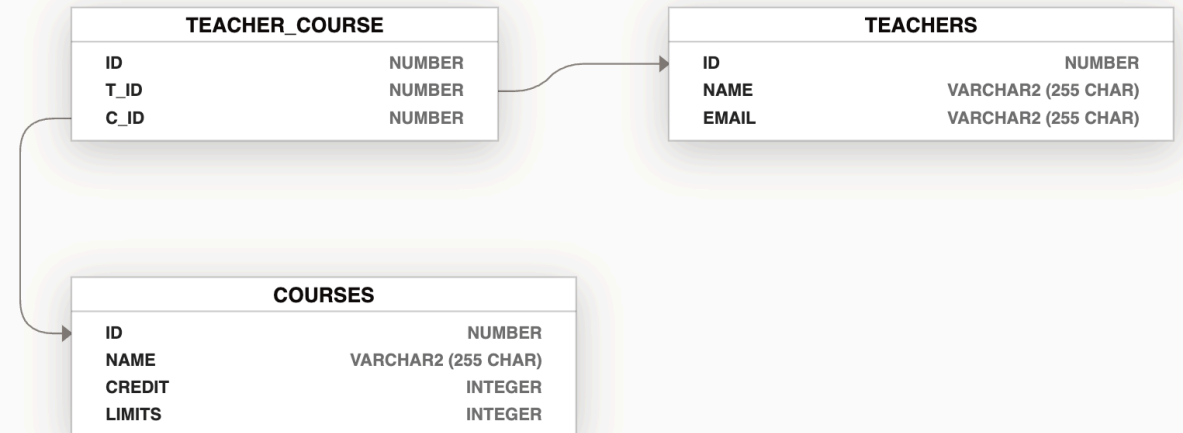
- 無法用階層式表格表示法建立多對多關係。
- 需要手動建立中間表格, 並在中間表格中建立兩個外鍵參考兩個表格的主鍵。

例如: 一位教師可以教多門課, 一門課
可有多位老師

Teachers
name
email

Courses
name
credit int [學分數]
limits int [人數上限]

Teacher_Course
t_id /fk Teachers [外鍵 Teachers]
c_id /fk Courses [外鍵 Courses]



Quick SQL

Help

Settings



SQL

Diagram

```
1 Teachers
2   name
3   email
4
5 Courses
6   name
7   credit int [學分數]
8   limits int [人數上限]
9
10 Teacher_Course
11   t_id /fk Teachers [外鍵 Teachers ]
12   c_id /fk Courses [外鍵 Courses]
13
```

```
1 -- create tables
2
3 create table Teachers (
4     id      number generated by default on null as identity
5     |      | constraint Teachers_id_pk primary key,
6     name    varchar2(255 char),
7     email   varchar2(255 char)
8 );
9
10
11
12 create table Courses (
13     id      number generated by default on null as identity
14     |      | constraint Courses_id_pk primary key,
15     name    varchar2(255 char),
16     credit  integer,
17     limits  integer
18 );
19
20 comment on column Courses.credit is '學分數';
21 comment on column Courses.limits is '人數上限';
22
23
24 create table Teacher_Course (
25     id      number generated by default on null as identity
26     |      | constraint Teacher_Course_id_pk primary key,
27     t_id    number
28     |      | constraint Teacher_Course_t_id_fk
29     |      | references Teachers,
30     c_id    number
31     |      | constraint Teacher_Course_c_id_fk
32     |      | references Courses
33 );
34
```

5. DDL 產生選項

產生 DDL 時, 可以指定選項, 方便程式開發:

1. 期望在表格名稱前加上特定的前綴字元, 例如: `app_` 或 `hr_` 以區分表格所屬的應用程式。
2. 產生 DROP TABLE 語句, 先刪除表格再建立新的表格, 避免重複建立表格。
3. 自動加入記錄的稽核欄位, 例如: `created_by`, `created_date`, `updated_by`, `updated_date`。

按 (B)Settings, 可以設定這些選項。

The screenshot shows the QuickSQL application interface. At the top, there is a menu bar with 'Help', 'Settings', and a vertical ellipsis icon. Below the menu bar, there are tabs for 'SQL' and 'Diagram'. The 'Settings' dialog box is open, displaying various configuration options. The dialog has a title bar with the word 'Settings' and a close button. The settings are organized into two main sections: 'Table' and 'Primary Key'. The 'Table' section includes fields for 'Object Prefix' and 'Schema', and toggle switches for 'Compression' (currently off) and 'Generate Inserts' (currently on). The 'Primary Key' section includes a toggle for 'Add Primary Key' (currently on), a dropdown menu for 'Population Method' (currently set to 'Identity Column'), and a partially visible toggle for 'Prefix primary keys with table name'. At the bottom of the dialog, there are three buttons: 'Cancel', 'Reset', and 'Save Changes'. Numbered callouts 1 and 2 are present: callout 1 points to the 'Settings' button in the main window's menu bar, and callout 2 points to the 'Settings' title bar of the dialog box.

Settings

Table

Object Prefix ?

Schema ?

Compression ☐ ?

Generate Inserts ☒ ?

Primary Key

Add Primary Key ☒ ?

Population Method ?

Prefix primary keys with table name ☐ ?

Cancel Reset Save Changes

6. 產生隨機資料

- 使用表格指令 `/insert n` 隨機產生 n 筆資料。
- 使用欄位指令 `/values` 來指定隨機產生的值。

Example: 產生 2 筆課程資料, 人數上限(limits)為 10, 20, 30 之一。

```
Courses /insert 2
  name
  credit int [學分數]
  limits int /values 10, 20, 30 [人數上限]
```

- 系統會自動產生 `INSERT INTO` 語句
- 並且重新設定 ID 產生器的起始值。
 - 上述例子中, 會將 ID 產生器的起始值設定為 3。

↑ Utilities \ Quick SQL

Quick SQL Help Settings ⋮ SQL Diagram

```
1 Teachers /insert 2
2   name
3   email
4
5 Courses /insert 2
6   name
7   credit int [學分數]
8   limits int /values 10, 20, 30 [人數上限]
9
10 Teacher_Course /insert 4
11   t_id /fk Teachers [外鍵 Teachers ]
12   c_id /fk Courses [外鍵 Courses]
13
```

```
70
71 insert into Courses (
72     id,
73     name,
74     credit,
75     limits
76 ) values (
77     1,
78     'Bessie Weaver',
79     29,
80     20
81 );
82 insert into Courses (
83     id,
84     name,
85     credit,
86     limits
87 ) values (
88     2,
89     'Sean Stevenson',
90     87,
91     10
92 );
93
94 commit;
95
96 alter table Courses
97 modify id generated always as identity restart start with 3;
98
```

測試範例

```
Teachers /insert 2  
  name  
  email
```

```
Courses /insert 2  
  name  
  credit int [學分數]  
  limits int /values 10, 20, 30 [人數上限]
```

```
Teacher_Course /insert 4  
  t_id /fk Teachers [外鍵 Teachers ]  
  c_id /fk Courses [外鍵 Courses]
```


7. 更多資訊

- Help in the Oracle Apex

The screenshot shows the Oracle APEX Quick SQL interface. The top navigation bar includes 'ORACLE APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. The 'SQL Workshop' tab is active. Below the navigation bar, the 'Quick SQL' section is visible, with buttons for 'Clear', 'Samples', 'Help' (highlighted with a yellow box), 'Generate SQL', 'SQL', 'Settings', 'Download', 'Save SQL Script', and 'Review and Run'. The 'Help' modal is open, displaying the 'About' tab. The modal content includes links for 'About Quick SQL', 'Getting Started', and 'Creating the Database Objects'. The 'About Quick SQL' section describes the tool's purpose and provides a list of top use cases.

Help

About Data Types Table Directives Column Directives Views Settings Samples

About

[About Quick SQL](#)
[Getting Started](#)
[Creating the Database Objects](#)

About Quick SQL

Quick SQL provides a quick way to generate the SQL required to create a relational data model from an indented text document. This tool is designed to reduce the time and effort required to create SQL tables, triggers, and index structures. This tool is not designed to be a replacement for data modeling, it is simply a quick way to develop a script for simple tables and views. Once the SQL is generated it can be tweaked and expanded upon.

Top Use Cases:

- Quickly create robust data models
- Easily generate random data
- Learn SQL create table, select, insert, index, trigger, PL/SQL package, and view syntax using provided examples

Getting Started

- Other resources
 - [Oracle APEX Tutorial 7 - Intro to Quick SQ](#)
 - [Oracle APEX Tutorial 8 - Quick SQL INSERT and Views](#)
 - [4.3 Using Quick SQL, Application Express SQL Workshop Guide](#)