

資料表之建立與修改

Structural Query Language

Data Definition Language

Data Manipulation Language

SQL Introduction

- SQL – Structural Query Language
 - DDL – Data Definition Language
 - DML – Data Manipulation Language
 - DCL – Data Control Language

Command	Description
SELECT	資料查詢
CREATE, ALTER, DROP RENAME, TRUNCATE	資料欄位屬性定義 (DDL)
INSERT, UPDATE DELETE, MERGE	資料增加、刪除、 修改、合併等處理(DML)
COMMIT、ROLLBACK SAVEPOINT	資料交易存取控制 (DML)
GRANT, REVOKE, AUDIT	使用權限稽核控管(DCL)

基本資訊查詢

- `SQL> conn username@dbname/password ;`
 - 連線至資料庫中之使用者綱要
 - `dbname`: 資料庫名稱
 - `username`: 使用者帳號
 - `password`: 使用者密碼
- `SQL> password`
 - 改變使用者密碼
- `SQL> SELECT table_name FROM user_tables;`
 - 查詢使用者綱要下所有表格資訊
- `SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'dd-mon-yyyy hh12:mi:ss pm';`
 - 將系統時間設為12時制之'日-月-西元年 時:分:秒 上或下午'
- `SQL> SELECT SYSDATE FROM DUAL;`
 - 查詢系統時間
- `SQL> SELECT (1+2)*4/1.5-3 FROM DUAL;`
 - 進行簡單的四則運算，先做括號內運算，再乘除，後加減

建立表格欄位

- CREATE TABLE *table_name*
(*col_name data_type* [DEFAULT *expr*], [...]);
 - *table_name*: 表格名稱
 - *col_name*: 欄位名稱
 - 表格及欄位名稱不可以數字為字首, 且不可為Oracle保留字
 - 名稱內之字元可為A-Z, a-z, 0-9, _, #, !
 - 同一個使用者綱要內不可有重複名稱之表格
 - *data_type*: 資料型態
 - CHAR(*n*): 最大為*n*個字元的固定長度字串, $n \leq 2000$
 - VARCHAR(*n*): 最大為*n*個字元的可變長度字串, $n \leq 2000$
 - VARCHAR2(*n*): 最大為*n*個字元的可變長度字串, $n \leq 4000$
 - NUMBER(*o*,*d*): 數字型態, $0 \leq o \leq 38$, $-84 \leq d \leq 127$
 - 為所有位數, *d*為小數點後位數;
 - d*省略時表整數, *o*,*d*均省略則為38位整數;
 - d*可為負值, 表四捨五入至整數之位數
 - DATE: 時間型態, 常與TO_DATE函式合用, 如
TO_DATE('2003-05-29', 'yyyy-mm-dd')
 - DEFAULT *expr*: 設定*expr*資料為預設值, 可省略
- DROP TABLE *table_name*
 - 刪除表格

輸入表格資料

- `INSERT INTO table_name`
`[(col1, col2, ...)] VALUES`
`(col1_value, col2_value, ...);`
 - `table_name`: 資料表名稱
 - `col1, col2, ...`: 欲輸入資料的欄位, 如省略則代表所有欄位均須輸入資料
 - `col1_value, col2_value, ...`: 對應欄位格式之資料輸入值
 - 如指定部分欄位輸入, 未列出之欄位會被自動輸入空值
 - 如對應之欄位有設預設值, 則輸入`default`可自動填入預設資料
 - 輸入日期格式時, 可配合`TO_DATE`函式之使用, 如
`TO_DATE('2003-05-29', 'yyyy-mm-dd')`
 - 輸入文字資料時, 須以單引號將資料括起來, 如'台中市'
 - 如對應之欄位無資料, 可輸入`null`或以單引號'' 取代
- `SELECT * FROM table_name;`
 - 查詢表格內所有資料
- `DESC table_name`
 - 查詢表格`table_name`之欄位資訊

- SQL> CREATE TABLE dept
2 (deptno NUMBER(4),
3 dname VARCHAR2(15),
4 loc VARCHAR2(20),
5 sdate DATE DEFAULT sysdate);
 - 建立部門dept資料表
 - deptno: 部門編號, 至多4碼整數
 - dname: 部門名稱, 至多15字元
 - loc: 部門所在地區, 至多20字元
 - sdate: 部門成立時間, 預設值為系統建表時間
- SQL> INSERT INTO dept VALUES
2 (101, 'MIS', 'Taichung', default);
 - 輸入含預設值之資料
- SQL> INSERT INTO dept VALUES
2 (102, 'HIS', 'Taipei',
3 TO_DATE('1999-03-23', 'yyyy-mm-dd'));
 - 使用TO_DATE函式輸入時間資料
- SQL> INSERT INTO dept (deptno, dname, sdate)
2 VALUES(101, '', null);
 - 指定欄位輸入含空值之資料

複製表格資料

- `CREATE TABLE tar_table AS
SELECT * FROM src_table;`
 - 將表格資料(含欄位)完全複製至新表格
 - `tar_table`: 新表格名稱
 - `src_table`: 原始表格名稱
- `INSERT INTO tar_table
[(tar_col1[, tar_col2, ...])]
SELECT src_col1[, src_col2, ...]
FROM src_table;`
 - 將表格資料部份匯入至新表格
 - 新表格須存在且建立好資料型態相符之欄位
 - `tar_col1, tar_col2, ...`: 新表格欄位名稱
 - `src_col1, tar_col2, ...`: 原始表格欄位名稱
 - 如省略新表格欄位名稱，可以星號 `*` 代表原始表格之所有欄位

查詢表格資料

- `SELECT * FROM table_name ;`
- `SELECT col1 [, col2, ...]`
`FROM table_name ;`
 - `table_name`: 表格名稱
 - `col1, col2, ...`: 欄位名稱，如以*取代則表示列出所有資料
- `SQL> CREA TABLE emp AS`
`2 SELECT * FROM scott.emp ;`
 - 自scott使用者綱要之emp表格複製所有資料至新建之emp資料表
- `SQL> INSERT INTO dept(deptno,dname,loc)`
`2 SELECT deptno,dname,loc`
`3 FROM scott.dept ;`
 - 將scott使用者者綱要之dept表格內deptno,dname,loc等欄位之資料複製到已存在之dept表內
- `SQL> SELECT dname, loc FROM dept ;`
 - 查詢dept表內的部門名稱及所在地區

修改表格欄位

- ALTER TABLE *table_name*
ADD (*col_name data_type* [DEFAULT *expr*]
[, ...]);
 - 增加新欄位，可一次增加數個欄位
 - *col_name*: 欄位名稱；*datatype*: 資料型態
 - DEFAULT *expr*: 設定*expr*資料為預設值，可省略
- ALTER TABLE *table_name*
MODIFY (*col_name data_type* [DEFAULT *expr*]
[, ...]);
 - 修改欄位屬性，可一次修改數個欄位
- ALTER TABLE *table_name*
DROP COLUMN *col_name* ;
 - 刪除欄位，一次刪除一個欄位
- ALTER TABLE *table_name*
RENAME COLUMN *col_name* TO *col_newname* ;
 - 欄位重新命名，一次更新一個欄位名稱
 - *col_newname*: 更名後之欄位名稱

- SQL> alter table dept
2 add (phone varchar2(10),
3 address varchar2(50)) ;
 - 將原部門資料表新增部門電話與地址之欄位
- SQL> alter table dept
2 modify (phone varchar2(12),
3 address number(5)) ;
 - 修改部門電話與位置之資料型態或格式
 - 修改資料格式時，如已有資料存在，則只允許增大限制範圍
- SQL> alter table dept
2 drop column phone ;
 - 刪除部門電話欄位
- SQL> alter table dept
2 rename column address to zip ;
 - 重新命名部門地址欄位為郵遞區號
- 原表格中既有資料會針對新增欄位以空值取代。

修改表格資料

- `UPDATE table_name`
`SET col_name = new_data`
`[, col_name = new_data,]`
`[WHERE conditions];`
 - 異動指定欄位之資料
 - `table_name`: 表格名稱
 - `col_name`: 欲異動資料之欄位名稱
 - `new_data`: 異動後之資料
 - `conditions`: 允許資料異動之條件
 - 異動後之資料必須符合設定之資料型態
 - 如果省略WHERE命令，該欄位所有資料都會被異動成新的資料
- `DELETE FROM table_name`
`[WHERE conditions];`
 - 刪除指定欄位中符合`conditions`條件之資料
 - 如果省略WHERE命令，則表格內所有資料會被刪除
- `COMMIT;`
- `ROLLBACK;`
 - 確認 (COMMIT) 或復原 (ROLLBACK) 前面修改資料的動作

- SQL> UPDATE dept
 - 2 SET deptno = 50,
 - 3 sdate = TO_DATE('1998-07-11', 'yyyy-mm-dd')
 - 4 WHERE deptno = 10;
- SQL> UPDATE dept SET sdate = sysdate
 - 2 WHERE deptno = 10
 - 3 OR dname = 'SALES';
- SQL> UPDATE dept
 - 2 SET loc = 'TAICHUNG'
 - 3 WHERE dname = 'ACCOUNTING'
 - 4 AND dname = 'FINANCE' ;
- SQL> COMMIT;
 - 確認以上之修改動作
- SQL> DELETE FROM dept
 - 2 WHERE deptno = 50;
- SQL> ROLLBACK;
 - 復原以上之刪除動作

範例資料

■ emp - 員工資料表欄位說明

Name	Null?	Type	說明
EMPNO	NOT NULL	NUMBER(4)	員工編號
ENAME		VARCHAR2(10)	名字
JOB		VARCHAR2(9)	職稱
MGR		NUMBER(4)	主管編號
HIREDATE		DATE	聘僱日期
SAL		NUMBER(7,2)	薪資
COMM		NUMBER(7,2)	獎金
DEPTNO		NUMBER(2)	部門編號

■ dept - 部門資料表欄位說明

Name	Null?	Type	說明
DEPTNO	NOT NULL	NUMBER(2)	部門編號
DNAME		VARCHAR2(14)	名稱
LOC		VARCHAR2(13)	所在地區

■ salgrade - 薪資等級資料表欄位說明

Name	Null?	Type	說明
GRADE		NUMBER	薪資等級
LOSAL		NUMBER	等級下限
HISAL		NUMBER	等級上限

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

基本查詢

- `SELECT [DISTINCT] col1 [, col2, ...]`
`FROM table_name`
`[WHERE conditions]`
`[ORDER BY col_order [DESC]];`
 - `DISTINCT`: 不列出重複資料，如省略則不過濾重複之資料
 - `col1, col2, ...`: 欄位名稱，如以*取代則表示列出所有資料
 - `table_name`: 表格名稱
 - `WHERE`: 加入查詢控制條件，如省略則列出指定欄位之所有資料
 - `conditions`: 查詢控制條件
 - 基本語法如：欄位名稱 = 資料值
資料如為字串則須以單引號括起來，且英文字元大小寫有分別
 - 如資料為數值或日期，可使用基本運算子如
`=, >, <, <=, >=, <>, !=`
 - 可用邏輯運算子如`AND, OR`及`NOT`組合多種控制條件
 - `ORDER BY col_order`: 以`col_order`欄位作遞增排序
 - 如加上`DESC`則表示作遞減排序，排序參考欄位可不只一個

■ Examples

- SQL> SELECT DISTINCT * FROM dept ;
 - 列出dept表內所有欄位不重覆之資料
- SQL> SELECT * FROM dept
2 WHERE deptno = 10;
 - 查詢dept表內部門編號為10的所有欄位資料
- SQL> SELECT * FROM dept
2 WHERE deptno > 10;
3 ORDER BY dname DESC;
 - 查詢dept表內部門編號大於10的所有欄位資料並依部門名稱做遞減排序
- SQL> SELECT dname, loc FROM dept
2 WHERE deptno = 10
3 AND NOT deptno < 30
4 OR loc != 'CHICAGO';
 - 查詢dept表內部門編號不小於30及等於10或不位於Chicago的部門名稱及所在地區

建立查詢欄位

- `SELECT col|expr [alias][, ...]`
`FROM table_name;`
 - 建立查詢欄位之別名
 - `col`: 欄位名稱
 - `expr`: 欄位運算
 - `alias`: 欄位別名
- Examples:
 - `SQL> SELECT ename, sal, sal*12 annual FROM emp;`
 - 將員工月薪*12後新增年收入欄位`annual`，但未修改原資料表
 - 如未建立別名，則以運算式作為查詢欄位名稱
 - `SQL> SELECT '(' || job || ') ' || ename "title - name" FROM emp ;`
 - 將員工職稱與名字結合成新欄位`title - name`，因新欄位中有空格，故須以雙引號括起來

查詢控制條件

- BETWEEN value1 AND value2
 - 查詢介於兩個控制值之間的數字、文字或日期資料
 - value1, value2: 數字、字串或日期
- Examples:
 - SQL> SELECT ename FROM emp
2 WHERE empno BETWEEN 7500 AND 7800;
 - 將員工編號介於7500和7800間之員工名字列出
 - SQL> SELECT ename from emp
2 WHERE ename BETWEEN 'ALICE' AND 'KING' ;
 - 將員工姓名介於ALICE和KING間之員工名字列出
 - SQL> SELECT ename from emp
2 WHERE hiredate
3 BETWEEN to_date('1981-01-01', 'yyyy-mm-dd')
4 AND to_date('1981-12-31', 'yyyy-mm-dd') ;
 - 將1981年1月1日到12月31日間僱用的員工名字列出

- IN (value1, value2, ...)
 - 查詢於括號集合內有列出的數字、文字或日期資料
 - value1, value2: 數字、字串或日期
- IS NULL
 - 查詢無資料之欄位
- Examples:
 - SQL> SELECT ename FROM emp
2 WHERE empno IN(7900,7902,7932);
 - 將員工編號為7900,7902和7932之員工名字列出
 - SQL> SELECT empno FROM emp
2 WHERE ename IN('ALLEN','ALICE','KING') ;
 - 將Allen,Alice,King三位員工之員工編號列出
 - SQL> SELECT ename FROM emp
2 WHERE hiredate IN(
3 TO_DATE('1981-12-01','yyyy-mm-dd'),
4 TO_DATE('1981-12-03','yyyy-mm-dd'),
5 TO_DATE('1981-12-06','yyyy-mm-dd')) ;
 - 將1981年12月1,3,6日3天顧用的員工名字列出
 - SQL> SELECT ename FROM emp
2 WHERE comm IS NULL ;
 - 找出沒有績效獎金的員工

■ LIKE *partial_value*

- 利用萬用字元%查詢包含指定字串的資料
- *partial_value*: 指定字串，例如
 - ' %s' : 以s結尾的字串
 - ' s%' : 以s開頭的字串
 - ' %s%' : 中間存在s的字串
 - ' _s%' : 第二個字元為s的字串
- 如果欲查出帶有保留字元(如_)的字串，則在保留字元前加上以ESCAPE定義之跳脫字元

■ Examples:

- SQL> SELECT ename FROM emp
2 WHERE ename LIKE 'S%';
 - 將名字以s開頭的之員工名字列出
- SQL> SELECT ename FROM emp
2 WHERE ename LIKE '%MAN%';
 - 將名字中有MAN之員工名字列出
- SQL> SELECT job FROM emp
2 WHERE job LIKE '_LERK';
 - 列出第二個字元起為LERK的職務名稱
- SQL> SELECT job FROM emp
2 WHERE job LIKE '%_%' ESCAPE '\';
 - 定義反斜線為跳脫字元以找出中帶有底線的職務名稱