關聯式資料查詢

Functions and Group

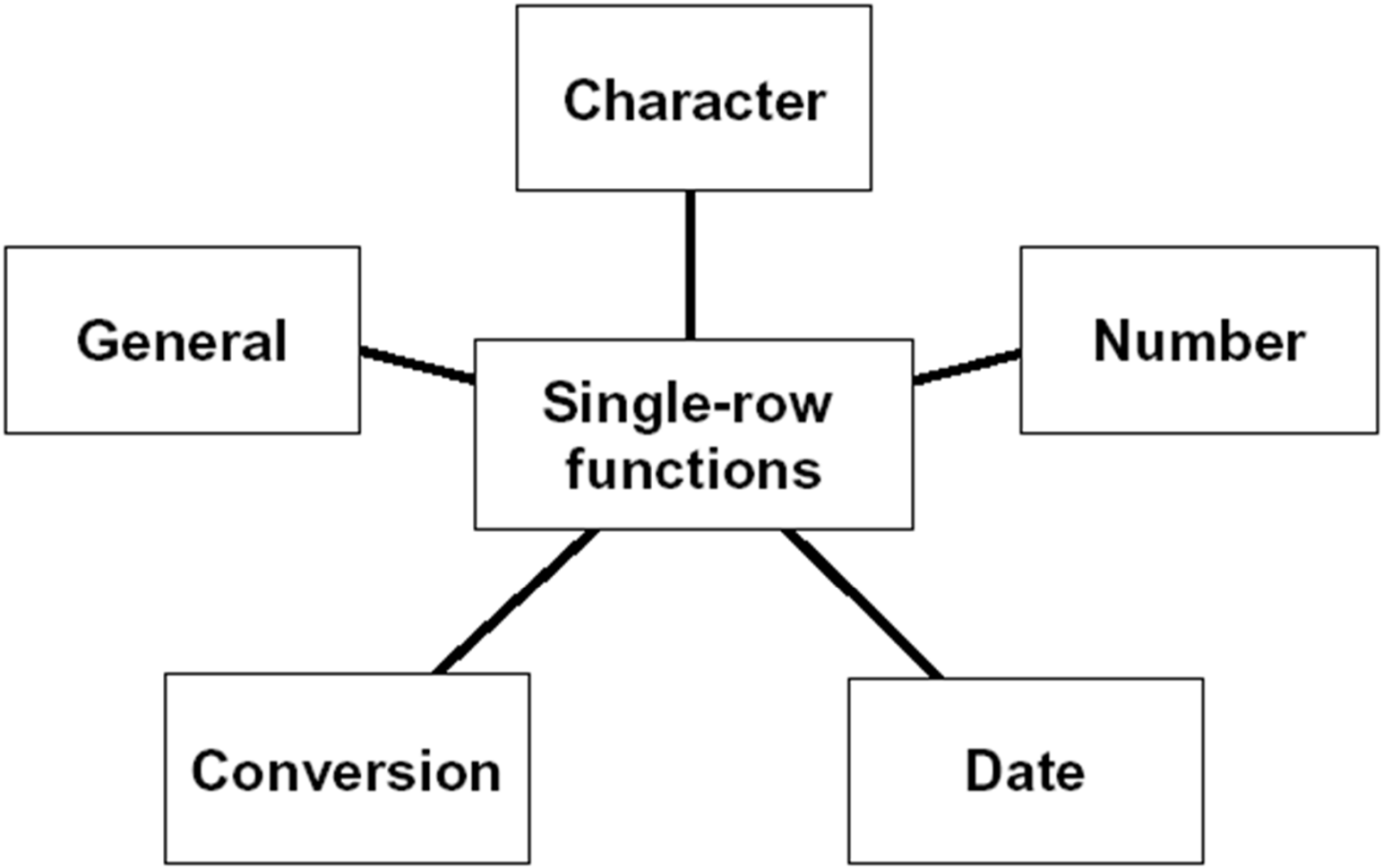
Sub Query

Join Data

View

內建函式介紹

- Oracle內建函式分類



- Character Function
  - SQL> SELECT LOWER('Hello World!') FROM DUAL ;
  - SQL> SELECT UPPER('Hello World!') FROM DUAL ;
  - SQL> SELECT INITCAP('Hello World!') FROM DUAL ;
  - SQL> SELECT CONCAT('Hello', ' World!') FROM DUAL ;
  - SQL> SELECT SUBSTR('Hello World!', 1, 5) FROM DUAL ;
  - SQL> SELECT SUBSTR('Hello World!', -5, 3) FROM DUAL ;
  - SQL> SELECT LENGTH('Hello World!') FROM DUAL ;
  - SQL> SELECT INSTR('Hello World!', 'W') FROM DUAL ;
  - SQL> SELECT TRIM('H' FROM 'Hello World!') FROM DUAL ;
  - SQL> SELECT LPAD(sal, 10, '*') FROM emp ;
  - SQL> SELECT RPAD(sal, 10, '*') FROM emp ;
  - SQL> SELECT ename FROM emp
    2 WHERE LOWER(ename) = 'smith' ;
  - SQL> SELECT empno,CONCAT(CONCAT(job,'-'),ename) NAME,
    2 LENGTH(ename),INSTR(ename,'A') "Contains 'A '?“
    3 FROM emp
    4 WHERE SUBSTR(ename, -1, 1) = 'N' ;

- Number Function
  - SQL> SELECT ROUND(45.923,2), ROUND(45.923,0), ROUND(45.923,-1)
    2 FROM DUAL ;
  - SQL> SELECT TRUNC(45.923,2), TRUNC(45.923), TRUNC(45.923,-2)
    2 FROM DUAL ;
  - SQL> SELECT ename, sal, MOD(sal, 5000)
    2 FROM emp
    3 WHERE job LIKE 'SA%' ;
  - Others: ABS, ACOS, ASIN, ATAN, ATAN2, CEIL, COS, COSH, EXP, FLOOR, LN, LOG, POWER, SIGN, SIN, SINH, SQRT, TAN, TANH,…etc.

- Date Function
  - Use ALTER SESSION to change time format, for example
    - ALTER SESSION SET NLS_DATE_FORMAT = ‘DD-Mon-YYYY hh24:mi’ ;
    - ALTER SESSION SET NLS_DATE_FORMAT = ‘DD-Mon-YYYY hh12:mi AM’ ;
  - SYSDATE – 傳回系統時間
    - SQL> SELECT sysdate, sysdate+5, sysdate-20 FROM dual ;
    - SQL> SELECT to_date('16-Mar-2005','dd-Mon-yyyy') - to_date('01-Mar-2005','dd-Mon-yyyy') FROM dual ;
    - SQL> SELECT sysdate + 8/24 FROM DUAL ;
    - SQL> SELECT ename, (SYSDATE - hiredate)/7 AS weeks
      2 FROM scott.emp
      3 WHERE deptno = 10 ;

- Date Function (continued)
  - MONTHS_BETWEEN – 計算間隔月數
    - SQL> SELECT MONTHS_BETWEEN ('01-SEP-2005','11-JAN-2010')
      2 FROM DUAL ;
  - NEXT_DAY – 計算某日的下一個星期日數
    - SQL> SELECT NEXT_DAY ('01-SEP-2005','FRIDAY') FROM DUAL ;
  - ADD_MONTHS – 計算某日幾個月後的日期
    - SQL> SELECT ADD_MONTHS ('11-JAN-2005',6) FROM DUAL ;
  - LAST_DAY – 傳回某日所屬月份的最後一日
    - SQL> SELECT LAST_DAY('01-FEB-2004') FROM DUAL ;
  - Others:
    - CURRENT_DATE, EXTRACT(datetime),ROUND(date),…, etc.

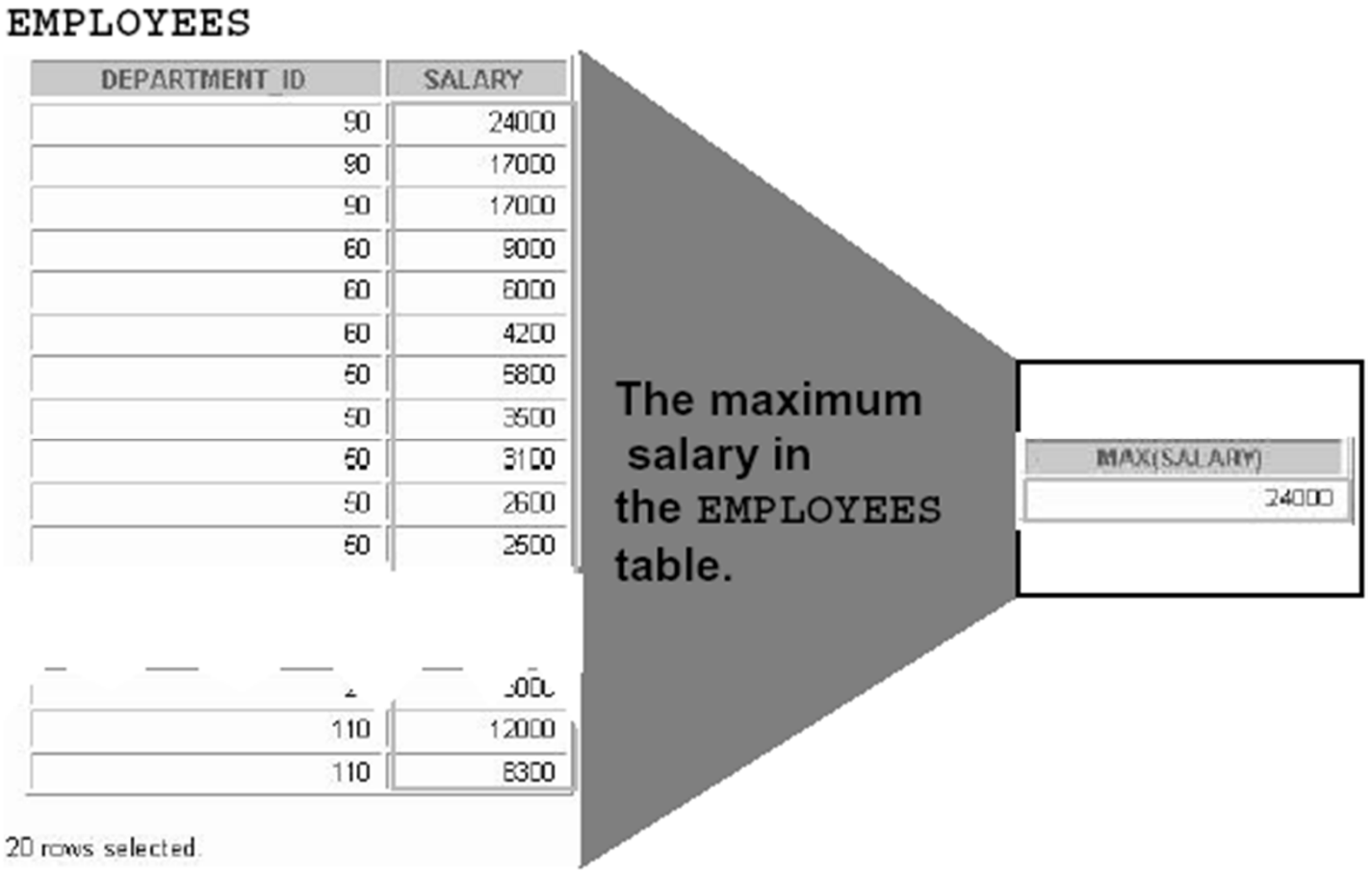
- Conversion Function
  - Date to Character
    - SQL> SELECT ename, TO_CHAR(hiredate, 'MM/YY') Month_Hired
      2 FROM emp ;
    - SQL> SELECT TO_CHAR(sysdate, 'Year-Mon-DD-Day-W-Q-A.D.') FROM DUAL ;
    - SQL> SELECT TO_CHAR(sal, '$99,999.00') sal
      2 FROM emp ;
  - Character to Date
    - SQL> SELECT ename, hiredate FROM emp
      2 WHERE hiredate = TO_DATE('May 01, 1981', 'Month DD, YYYY') ;
    - SQL> SELECT ename, hiredate FROM emp
      2 WHERE hiredate = TO_DATE('05-01-1981', 'MM-DD-YYYY') ;
  - Others:
    - TO_NUMBER, TO_CLOB, …, etc.
  - Date Format: (next slide)

Element	Description
SCC or CC	Century; server prefixes B.C. date with -
Years in dates YYYY or SYYYY	Year; server prefixes B.C. date with -
YYY or YY or Y	Last three, two, or one digits of year
Y,YYY	Year with comma in this position
IYYY, IYY, IY, I	Four, three, two, or one digit year based on the ISO standard
SYEAR or YEAR	Year spelled out; server prefixes B.C. date with -
BC or AD	B.C.A.D. indicator
B.C. or A.D.	B.C./A.D. indicator with periods
Q	Quarter of year
MM	Month: two-digit value
MONTH	Name of month padded with blanks to length of nine characters
MON	Name of month, three-letter abbreviation
RM	Roman numeral month
WW or W	Week of year or month
DDD or DD or D	Day of year, month, or week
DAY	Name of day padded with blanks to a length of nine characters
DY	Name of day; three-letter abbreviation
J	Julian day; the number of days since 31 December 4713 B.C.

- General Function
  - NVL(expr1, expr2)
    - SQL> SELECT comm, NVL(comm, 0)
      2 FROM emp ;
  - NVL2(expr1, expr2, expr3)
    - SQL> SELECT comm, NVL2(comm, 'up', 'down')
      2 FROM emp ;
  - NULLIF(expr1, expr2)
    - SQL> SELECT job, LENGTH(job) "expr1",
      2 ename, LENGTH(ename) "expr2",
      3 NULLIF(LENGTH(job), LENGTH(ename)) result
      4 FROM emp ;
  - COALESCE(expr1, expr2, ..., exprn)
    - SQL> SELECT comm, sal, COALESCE(comm, sal, 10)
      2 FROM scott.emp ;
  - Others:
    - CASE Selector, DECODE, EXTRACTVALUE, …, etc.

群組概念(GROUP)

- GROUP Functions: 將資料做分類統計計算
  - `SELECT [column,] group_function(column) FROM table WHERE condition] ;`



- COUNT – 計算總數
  - SQL> SELECT COUNT(empno) FROM emp;
- MAX, MIN – 計算最大或最小值
  - SQL> SELECT MIN(sal) FROM emp;
  - SQL> SELECT MAX(sal) FROM emp;
- SUM – 計算加總值
  - SQL> SELECT SUM(sal) FROM emp;
- AVG – 計算平均值
  - SQL> SELECT AVG(sal) FROM emp;
- 以群組函數作為控制條件
  - SQL> SELECT ename FROM emp
    2 WHERE sal = (SELECT MAX(sal) FROM emp) ;
    - 找出薪水最多的員工
- 配合其他函數使用
  - SQL> SELECT AVG(NVL(comm, 0))
    2 FROM emp ;
    - 算出員工平均紅利獎金
    - 如儲存格無資料(null)，則改為0再納入計算
    - 對於所有儲存格為空值的資料，群組函式均不會計算進去

整合查詢(Aggregation Query)

- 使用GROUP BY語法
  - SELECT [column,]
    group_function(column), …
    FROM table
    [WHERE condition]
    [GROUP BY group_by_expression]
    [HAVING group_by_condition]
    [ORDER BY column] ;
  - 與查詢欄位一起使用－利用GROUP BY語法使指定的欄位資料和群組資料數一致
  - SQL> SELECT deptno, MIN(sal)
      2 FROM emp
      3 GROUP BY deptno ;
    - 查出各部門的最低薪資
    - 將部門編號作群組分類，再把分類後的部門以群組函數計算

- HAVING－在群組中加入控制條件
  - SQL> SELECT deptno, MIN(sal)
    2 FROM emp
    3 GROUP BY deptno
    4 HAVING AVG(sal) > 1000 ;
    - 查詢平均薪資大於1000的部門和其最少薪資
    - HAVING語句中定義的條件式應使用群組函或用於GROUP BY的欄位
- 與WHERE條件控制一起查詢
  - SQL> SELECT deptno, COUNT(empno)
    2 FROM emp
    3 WHERE empno > 7500
    4 GROUP BY deptno
    5 HAVING COUNT(empno) > 3;
    - 查詢員工數目大於3人且員工編號在7500之後的部門
    - 群組函式的比較控制不可使用where語句
    - 使用WHERE時機在於將其他欄位資料加入條件控制
    - WHERE控制須在GROUP BY之前，HAVING必須在GROUP BY之後

- 排序整合查詢結果
  - SQL> SELECT deptno, COUNT(empno) "EMP Number"
    2 FROM emp
    3 GROUP BY deptno
    4 HAVING COUNT(empno) > 3
    5 ORDER BY "EMP Number" DESC;
    - 查詢員工數目大於3人的部門，並依員工數目遞減排序
    - 排序時使用查詢結果所列出之欄位，可使用別名及群組函數
    - 以HAVING作條件控制時不可使用別名
- 其他群組函數
  - STDDEV, STDDEV_POP, STDDEV_SAMP, VAR_POP, VAR_SAMP, VARIANCE, …, etc.
  - REGR_xxx表統計迴歸函式，如SLOPE, INTERCEPT, COUNT, R2, AVGX, AVGY, SXX, SYY, SXY

表格合併查詢

- JOIN觀念 – 利用表格關聯性，將多個表格合併成一個虛擬表格進行查詢
  - emp(empno,ename,…,deptno)
    dept(deptno,dname,loc)
    JOIN→(empno,ename,…..,deptno,dname,loc)
  - SELECT alias1.column1, alias2.column2, …
    FROM table1 alias1, table2 alias2, …
    [WHERE Clause] ;
  - SQL> SELECT empno,ename,sal,dname,loc
    2 FROM emp e, dept d
    3 WHERE e.deptno = d.deptno ;
    - 將員工資料表與部門資料表以部門編號(deptno)關聯做合併
    - 利用WHERE條件控制使關聯欄位合併兩個表格
    - 所列出的表格欄位如重複(例如deptno)，則須在欄位前面加上所屬表格名稱或別名
    - 如未在FROM語句中設定表格別名，則必須使用全名

- 使用JOIN語句
  - SELECT column1, column2, …
    FROM table1 JOIN table2
    ON table1.join_col = table2.join_col ;
  - SQL> SELECT empno,ename,sal,dname,loc
      2 FROM emp e JOIN dept d
      3 ON e.deptno = d.deptno ;
    - 將員工資料表與部門資料表以部門編號(deptno)關聯做合併
    - 利用JOIN…ON指定合併表格及欄位欄位
  - SELECT column1, column2, …
    FROM table1 NATURAL JOIN table2 ;
  - SQL> SELECT empno,ename,sal,dname,loc
      2 FROM emp NATURAL JOIN dept ;
    - 將員工資料表與部門資料表以部門編號(deptno)關聯做合併
    - 合併表格中同名之關聯欄位

- Noequijoin –
  - 合併表格中不存在具關聯鍵的欄位，但有關聯性
    - `emp(empno, ename, …,sal,…), salgrade(grade,losal,hisal)`

EMPLOYEES

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hunold	9000
Ernst	6000
Lorentz	4200
Mourgos	5800
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
ray	.00
Higgins	12000
Gietz	8300

20 rows selected.

JOB_GRADES

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

6 rows selected.

← **Salary in the `EMPLOYEES` table must be between lowest salary and highest salary in the `JOB_GRADES` table.**

```
SQL> SELECT ename,sal,grade
  2  FROM emp e, salgrade s
  3  WHERE e.sal BETWEEN s.losal AND
s.hisal ;
```

- 將員工資料表(emp)與薪資等級表(salgrade)合併後列出員工薪水等級

子查詢(Sub Query)

- Sub Query
  - 查詢中帶有其他的查詢，或利用查詢結果作為另一個查詢的控制條件
  - SELECT columns FROM tables WHERE Clause (SELECT columns FROM tables WHERE Clause);
  - SQL> SELECT ename, job
    2 FROM emp
    3 WHERE job = (SELECT job
    4 FROM emp
    5 WHERE empno = 7900)
    6 AND sal > (SELECT sal
    7 FROM emp
    8 WHERE empno = 7900) ;
    - 將員工資料表中職稱與編號7900員工相同，且薪資高於該員工的員工名字及其薪資列出

- 與表格合併觀念使用
  - SQL> SELECT ename, sal, dname
    2 FROM emp NATURAL JOIN dept
    3 WHERE sal > (SELECT AVG(sal) FROM emp) ;
    - 將員工資料表中薪資高於公司平均薪資的員工及其薪資、部門名稱列出
- 與群組分類合用
  - SQL> SELECT ename, sal, dname
    2 FROM emp NATURAL JOIN dept
    3 WHERE deptno IN
    4 (SELECT deptno FROM emp
    5 GROUP BY deptno
    6 HAVING AVG(sal) > 2000) ;
    - 將部門平均薪資大於2000的部門員工及員工薪資、部門名稱列出

- 在子查詢中使用空值觀念
  - SQL> SELECT ename FROM emp
    2 WHERE empno NOT IN
    3 (SELECT mgr FROM emp
    4 WHERE mgr IS NOT NULL) ;
    - 找出員工資料表中不是經理的員工(mgr欄位即代表該員工所屬之經理員工編號)
- 集合觀念－UNION, INTERSECT, MINUS
  - 聯集(UNION)
    SQL> SELECT ename FROM emp
    2 WHERE sal > 1000
    3 UNION
    4 SELECT ename FROM emp
    5 WHERE deptno > 10 ;
    - 將員工資料表中薪資大於1000與所屬部門編號大於10的員工聯集列出來
  - 同理可查詢交集(INTERSECT)與差集(MINUS)的結果

- 利用rownum虛擬欄位列出部分資料
  - ```
    SQL> SELECT rownum, ename, sal FROM (
      2     SELECT ename, sal FROM emp
      3     ORDER BY sal DESC)
      4  WHERE rownum<=5 ;
    ```
    - 找出薪水最多的前五名員工，列出姓名與薪水
    - 在第一個SELECT語法中的rownum目的在顯示排序的順位，如不呈現出來可以略去。
    - 子查詢中可得到被排序後的資料。
    - 在WHERE語句中的rownum用來控制顯示資料列數
- 利用子查詢更新資料
  - ```
    SQL> UPDATE dept
      2  SET loc='TAIPEI'
      3  WHERE deptno = (
      4     SELECT deptno FROM dept
      5     WHERE dname='OPERATIONS') ;
    ```
    - 將OPERATIONS部門的所在城市搬到TAIPEI
    - 在不知部門編號的情況下，利用子查詢找出OPERATIONS部門的編號，作為WHERE語句的控制條件。

視觀表(View)

- VIEW –
  - 將查詢結果儲存起來建立成報表，可視為一份查詢報告表格，執行相同的查詢時可直接查詢此報表的內容，以簡化複雜的查詢指令。
  - 當表格資料有異動時，報表資料自動更新。
  - CREATE [OR REPLACE] VIEW *view_name* (*col1, col2*, …) AS SELECT clause
    - 如重新命名欄位名稱，則必須與查詢語句所得到的欄位數相同
    - 如省略(…)內之欄位名稱，則自動以查詢語句所得到的欄位名稱命名
    - 欲修改VIEW的內容時，則加上OR REPLACE語句將查詢內容重新置換
  - DROP VIEW view_name ;
    - 刪除VIEW

- Example:
  - SQL> CREATE OR REPLACE VIEW v$emp_dept
    2 (eid, ename, income, dname) AS
    3 SELECT empno, ename, sal*12, dname
    4 FROM emp NATURAL JOIN dept ;
    - 將員工資料表與部門資料表合併後列出員工編號、員工名字、年收入及所屬部門名稱，並建立成VIEW
  - SQL> SELECT ename, income, dname FROM v$emp_dept ;
    - 從VIEW中列出員工名字、年收入及部門名稱
    - 當員工資料表有增加新員工時，此視觀表資料會隨之異動
  - SQL> CREATE VIEW v$saltop10 AS
    2 SELECT empno, ename, sal
    3 FROM (SELECT empno, ename, sal
    4 FROM emp ORDER BY sal DESC)
    5 WHERE rownum <= 10 ;
    - 建立一個VIEW列出員工資料表中薪資最高10名員工資料
    - rownum為系統預設於資料表中之虛擬欄位

權限控管概念

- 權限控管的意義
  - 基於資訊安全，資料庫管理系統設有權限層級，一般使用者無法獲得系統資訊。
  - 經過資料擁有者的授權，被授權者可以查詢、新增、修改或刪除授權者的資料。
- 授權 – 將table_n的使用權限賦予user_n，n表一次可授權多個
  `GRANT SELECT, [UPDATE], [INSERT], [DELETE], […] ON table_n TO user_n`
  - SQL> GRANT SELECT ON emp TO scott ;
    - 將emp表格的查詢權限壽與使用者scott
- 撤銷 – 將user_n使用table_n的權限移除，n表一次可撤銷多個
  `REVOKE SELECT, [UPDATE], [INSERT], [DELETE], […] ON table_n FROM user_n`
  - SQL> REVOKE UPDATE, DELETE ON emp FROM scott, snowball ;
    - 將使用者scott和snowball修改和刪除emp表格資料的權限撤銷