CASE Expression

- General form (use searching condition):

  Machines (serialno, type, year, hours_used, accidents)

- *Find the rate of the accidents of "chain saw" in the whole accidents :*

```
SELECT sum (CASE
                        WHEN type='chain saw' THEN accidents
                        ELSE 0e0
                END) / sum (accidents)
FROM Machines;
```

CASE Expression

- *Find the average accident rate of every kind of equipment :*

```
SELECT type, CASE
                        WHEN sum(hours_used)>0 THEN
                              sum(accidents)/sum(hours_used)
                        ELSE NULL
                  END AS accident_rate
FROM Machines
GROUP BY type;
```

(Because some equipments maybe not in use at all, their hours_used is 0. Use CASE can prevent the expression divided by 0.)

CASE Expression

- Compared with

```
SELECT type, sum(accidents)/sum(hours_used)
FROM Machines
GROUP BY type
HAVING sum(hours_used)>0;
```

Some New Features of SQL

- CAST expression
- CASE expression
- **Sub-query**
- Outer Join
- Recursion

Sub-query

- Embedded query & embedded query with correlation
- The functions of sub-queries have been enhanced in new SQL standard. Now they can be used in SELECT and FROM clause
  - Scalar sub-query
  - Table expression
  - Common table expression

Scalar Sub-query

- The result of a sub-query is a single value. It can be used in the place where a value can occur.
- *Find the departments whose average bonus is higher than average salary :*

```
SELECT d.deptname, d.location
FROM dept AS d
WHERE (SELECT avg(bonus)
           FORM emp
           WHERE deptno=d.deptno)
       > (SELECT avg(salary)
           FORM emp
           WHERE deptno=d.deptno)
```

Scalar Sub-query

- *List the deptno, deptname, and the max salary of all departments located in New York :*

```
SELECT d.deptno, d.deptname, (SELECT MAX (salary)
                              FROM emp
                              WHERE deptno=d.deptno) AS maxpay
FROM dept AS d
WHERE d.location = 'New York' ;
```

Table Expression

- The result of a sub-query is a table. It can be used in the place where a table can occur.

```
SELECT startyear, avg(pay)
FROM (SELECT name, salay+bonus AS pay,
                    year(startdate) AS startyear
          FROM emp) AS emp2
GROUP BY startyear;
```

- *Find departments whose total payment is greater than 200000*

```
SELECT deptno, totalpay
FROM (SELECT deptno, sum(salay)+sum(bonus) AS totalpay
          FROM emp
          GROUP BY deptno) AS payroll
WHERE totalpay>200000;
```

- Table expressions are temporary views in fact.