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.

Common Table Expression

- In some complex query, a table expression may need occurring more than one time in the same SQL statements. Although it is permitted, the efficiency is low and there maybe inconsistency problem.
- WITH clause can be used to define a common table expression. In fact, it defines a temporary view.
- *Find the department who has the highest total payment :*

Common Table Expression

- *Find the department who has the highest total payment :*

```
WITH payroll (deptno, totalpay) AS
        (SELECT deptno, sum(salary)+sum(bonus)
        FROM emp
        GROUP BY deptno)
SELECT deptno
FROM payroll
WHERE totalpay = (SELECT max(totalpay)
                         FROM payroll);
```

- Common table expression mainly used in queries which need multi level focuses.

Common Table Expression

- *Find department pairs, in which the first department's average salary is more than two times of the second one's :*

```
WITH deptavg (deptno, avgsal) AS
      (SELECT deptno, avg(salary)
       FROM emp
       GROUP BY deptno)
SELECT d1.deptno, d1.avgsal, d2.deptno, d2.avgsal
FROM deptavg AS d1, deptavg AS d2
WHERE d1.avgsal>2*d2.avgsal;
```

Some New Features of SQL

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

Outer Join

Teacher (name, rank)
Course (subject, enrollment, quarter, teacher)

```
WITH
    innerjoin(name, rank, subject, enrollment) AS
            (SELECT t.name, t.rank, c.subject, c.enrollment
             FROM teachers AS t, courses AS c
             WHERE t.name=c.teacher AND c.quarter=‘Fall 96’) ,
    teacher-only(name, rank) AS
            (SELECT name, rank
             FROM teachers
             EXCEPT ALL
             SELECT name, rank
             FROM innerjoin) ,
    course-only(subject, enrollment) AS
            (SELECT subject, enrollment
             FROM courses
             EXCEPT ALL
             SELECT subject, enrollment
             FROM innerjoin)
```

Outer Join

```
SELECT name, rank, subject, enrollment
FROM innerjoin
UNION ALL
SELECT name, rank,
        CAST (NULL AS Varchar(20)) AS subject,
        CAST (NULL AS Integer) AS enrollment
FROM teacher-only
UNION ALL
SELECT CAST (NULL AS Varchar(20)) AS name,
        CAST (NULL AS Varchar(20)) AS rank,
        subject, enrollment
FROM course-only ;
```