# Joins and Subqueries for two- and three-table joins:

# Comparing traditional and modern methods

# SELECT – Traditional Join 2 Tables

What are the
(1) SURNAMEs and (2) SALARY of current employees

```
SELECT e1.surname, j1.salary
FROM employee e1, jobhistory J1
WHERE e1.empno = j1.empno
AND j1.enddate IS NULL
```

# SELECT – Modern Join 2 Tables

What are the

(1) SURNAMEs and (2) SALARY of current employees

```
SELECT e1.surname, j1.salary
FROM employee e1 JOIN jobhistory J1
    ON e1.empno = j1.empno
WHERE  j1.enddate IS NULL
```

# SELECT – Traditional Join 3 Tables

What are the<br>
(1) SURNAMEs and<br>
(2) SALARY and<br>
(3) Department of current employees

```
SELECT e1.surname, j1.salary, dname
FROM  employee e1, jobhistory J1, department  D1

WHERE e1.empno = j1.empno
AND e1.depno = d1.depno
AND j1.enddate IS NULL
```

# SELECT – Modern Join 3 Tables

What are the

(1) SURNAMEs and

(2) SALARY and

(3) Department of current employees

```
SELECT e1.surname, j1.salary, dname
FROM  employee e1
JOIN jobhistory J1
    ON e1.empno = j1.empno
JOIN department  D1
    ON e1.depno = d1.depno
WHERE j1.enddate IS NULL
```

# More complex examples.

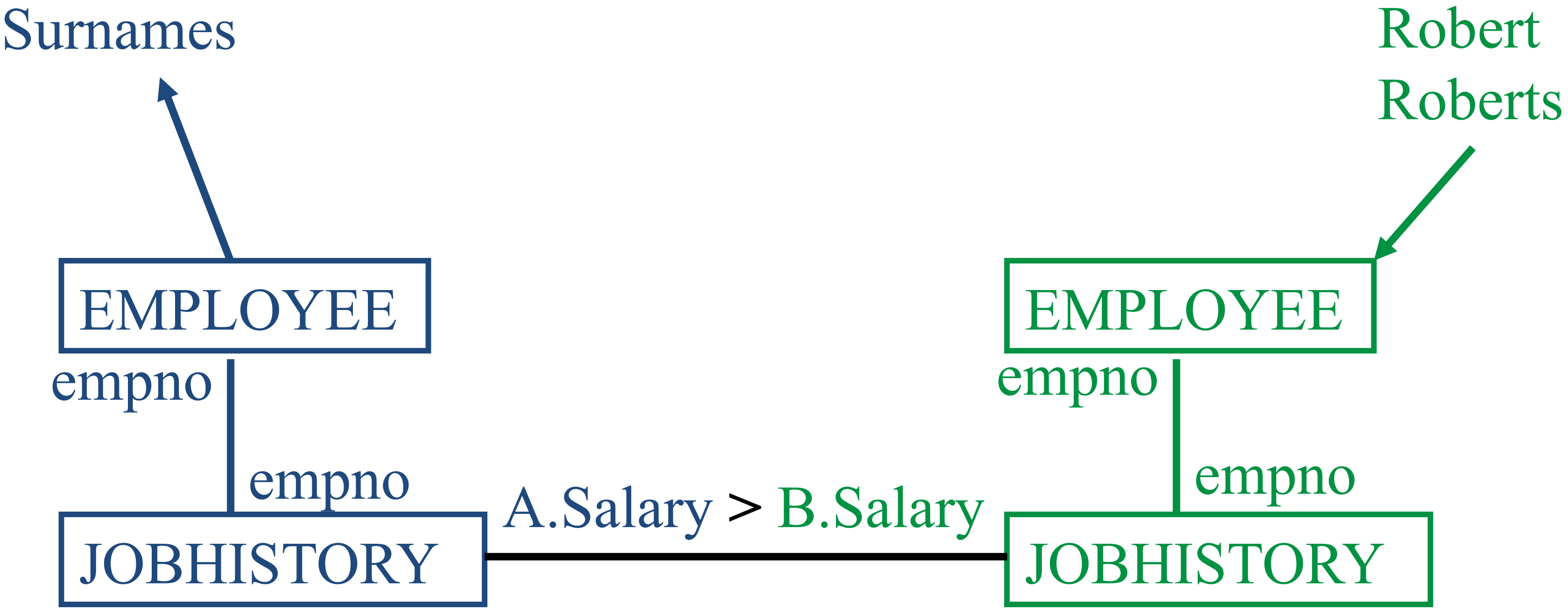
The next three SELECT statements are **equivalent**

(they give the same result).

What are the

(1) SURNAMEs and SALARY of employees who have a SALARY >

(2) the SALARY of Robert Roberts



# SELECT – Traditional Join

What are the

(1) SURNAMEs and SALARY of employees who have a SALARY **>**

(2) the SALARY of Robert Roberts

```
SELECT e1.surname, j1.salary
FROM  employee e1, jobhistory J1, jobhistory J2,
    employee e2
WHERE                          e1.empno = j1.empno AND
    j1.enddate IS NULL
AND e2.empno = j2.empno AND j2.enddate IS NULL
AND e2.surname = 'Roberts' AND e2.Forenames =
    'Robert‘
AND j1.salary > j2.salary
```

# SELECT – Modern Join

What are the

(1) SURNAMEs and SALARY of employees who have a SALARY **>**

(2) the SALARY of Robert Roberts

comma

```
SELECT e1.surname, j1.salary
FROM
employee e1 JOIN jobhistory J1 ON e1.empno =
     j1.empno,
 jobhistory J2 JOIN employee e2 ON e2.empno =
     j2.empno
WHERE j1.enddate IS NULL
AND e2.surname = 'Roberts' AND e2.Forenames =
     'Robert'
AND j2.enddate IS NULL
```

# SELECT – Subquery

What are the

(1) SURNAMEs and SALARY of employees who have a SALARY **>**

(2) the SALARY of Robert Roberts

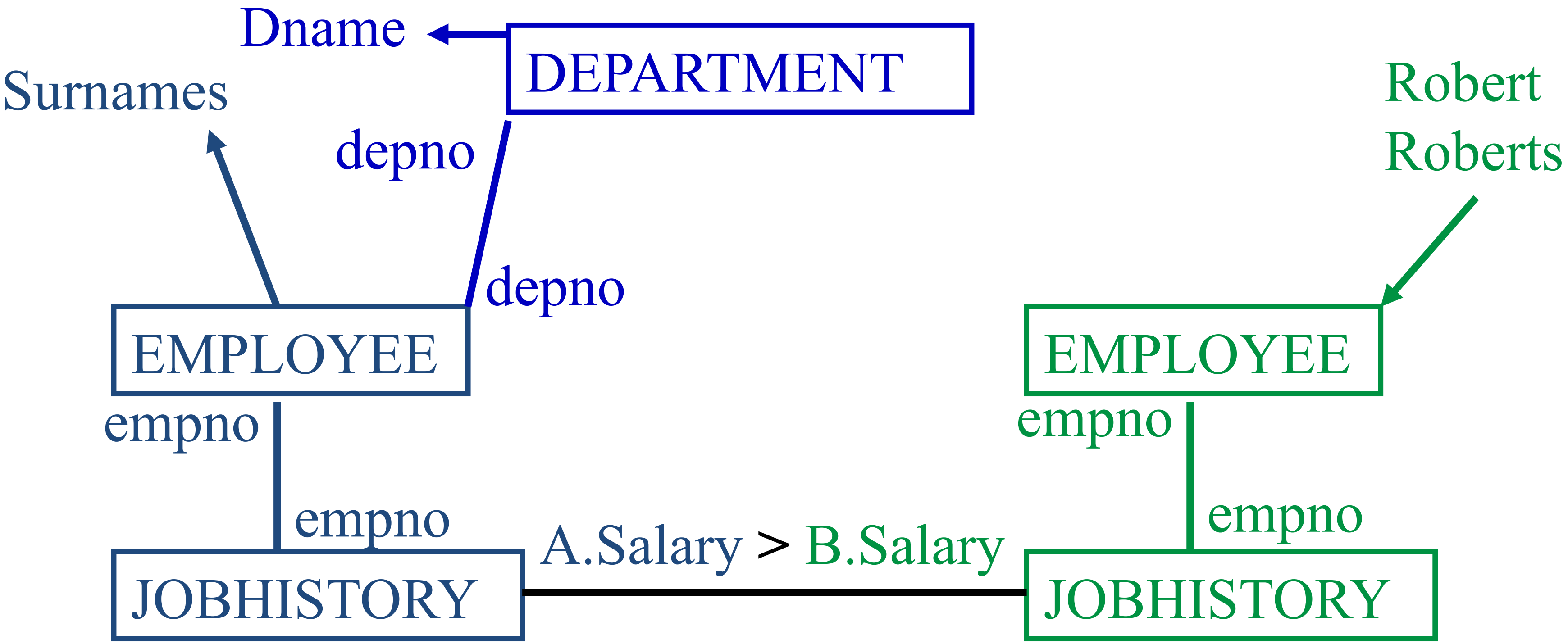
```
SELECT e1.surname, j1.salary
FROM  employee e1 JOIN jobhistory J1 ON e1.empno =
    j1.empno
WHERE                         j1.enddate IS NULL
AND j1.salary >
      ( SELECT j2.salary
      FROM jobhistory J2 JOIN employee e2 ON e2.empno =
    j2.empno
      WHERE e2.surname = 'Roberts'   AND e2.Forenames =
    'Robert'
      AND j2.enddate IS NULL )
```

# Adding a table to a complex query

What are the

(1) SURNAMEs and **DEPARTMENT** of employees who have a SALARY **>**

(2) the SALARY of Robert Roberts



# SELECT – Traditional - Adding a Table

What are the

(1) SURNAMEs and **DEPARTMENT** of employees who have a SALARY **>**

(2) the SALARY of Robert Roberts

comma

```
SELECT e1.surname, j1.salary, dname
FROM  employee e1, jobhistory J1, department as
    D1,
    jobhistory J2, employee e2
WHERE                          e1.empno = j1.empno
AND e1.depno = d1.depno
AND j1.enddate IS NULL
AND j1.salary > j2.salary
AND e2.empno = j2.empno
AND e2.surname = 'Roberts' AND e2.Forenames =
    'Robert'
AND j2.enddate IS NULL
```

# SELECT – Modern - Adding a Table

What are the

(1) SURNAMEs and **DEPARTMENT** of employees who have a SALARY **>**

(2) the SALARY of Robert Roberts

comma

```
SELECT e1.surname, j1.salary, dname
FROM  employee e1
JOIN jobhistory J1 ON e1.empno = j1.empno
JOIN department D1 ON e1.depno = d1.depno ,
 jobhistory J2 JOIN employee e2 ON e2.empno =
     j2.empno
WHERE j1.salary > j2.salary
AND e2.surname = 'Roberts' AND e2.Forenames =
     'Robert'
AND j2.enddate IS NULL AND j1.enddate IS NULL
```

# To exclude Robert Roberts, add this inside the subquery or at the end any of the other queries:

```
And e1.empno != e2.empno
```

(not actually needed here, as we select only people who earned **more** than him)