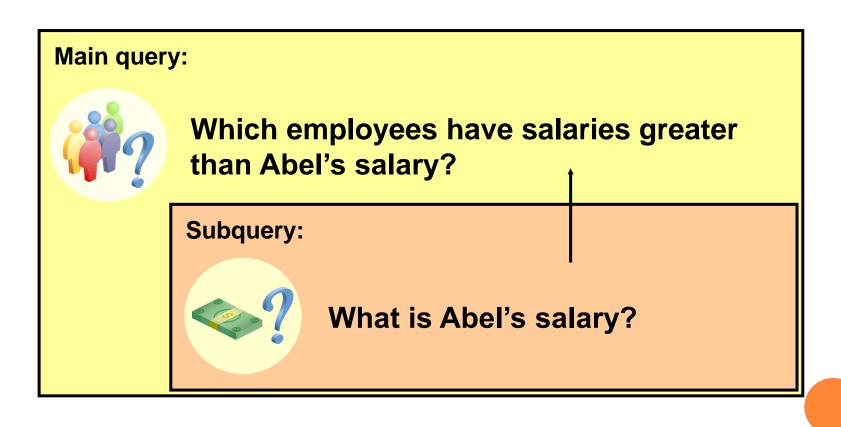
SUBQUERY

USING A SUBQUERY TO SOLVE A PROBLEM

• Who has a salary greater than Abel's?



SUBQUERY SYNTAX

```
SELECT select_list
FROM table
WHERE expr operator

(SELECT select_list
FROM table);
```

- The subquery (inner query) executes once before the main query (outer query).
- The result of the subquery is used by the main query.

Using a Subquery

Guidelines for Using Subqueries

- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison condition.
- Use single-row operators with single-row subqueries, and use multiple-row operators with multiple-row subqueries.

Types of Subqueries

- Single-row subquery
- Multiple-row subquery

SINGLE-ROW SUBQUERIES

- Return only one row
- Use single-row comparison operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

EXECUTING SINGLE-ROW SUBQUERIES

Using Group Functions in a Subquery

THE HAVING CLAUSE WITH SUBQUERIES

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) >

(SELECT MIN(salary)
FROM employees
WHERE department_id = 50);
```

WHAT IS WRONG WITH THIS STATEMENT?

```
ERROR at line 4:
ORA-01427: single-row subquery returns more than
one row
```

Single-row operator with multiple-row subquery

MULTIPLE-ROW SUBQUERIES

- Return more than one row
- Use multiple-row comparison operators

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

USING THE ANY OPERATOR IN MULTIPLE-ROW SUBQUERIES

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ANY

(SELECT salary
FROM employees
WHERE job_id = 'IT_PROG')</pre>
```

USING THE ALL OPERATOR IN MULTIPLE-ROW SUBQUERIES

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL

(SELECT salary
FROM employees
WHERE job_id = 'IT_PROG')</pre>
```