There are several ways to access multiple tables in the same query:

1. Sub-queries
2. Implicit JOIN
3. JOIN operators (INNER JOIN, OUTER JOIN, etc.)

In this lesson we will examine the first two options. The third is covered in more details in additional lessons.

1. Sub-queries

In a previous lesson we learned how to use sub-queries. Now let's use sub-queries to work with multiple tables. Lets consider the EMPLOYEES and department tables from the previous lesson.

EMPLOYEES:

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY
E1002	Alice	James	1234…	1972-07-31	F	980 Berry l…	200	80000.00
E1003	Steve	Wells	1234…	1980-08-10	M	291 Springs…	300	50000.00
E1004	Santosh	Kumar	1234…	1985-07-20	M	511 Aurora …	400	60000.00
E1005	Ahmed	Hussain	12341…	1981-01-04	M	216 Oak Tre…	500	70000.00
E1006	Nancy	Allen	12341…	1978-02-06	F	111 Green Pl…	600	90000.00
E1007	Mary	Thomas	12341…	1975-05-05	F	100 Rose Pl…	650	65000.00
E1008	Bharath	Gupta	12341…	1985-05-06	M	145 Berry L…	660	65000.00
E1009	Andrea	Jones	12341…	1990-07-09	F	120 Fall Cre…	234	70000.00
E1010	Ann	Jacob	12341…	1982-03-30	F	111 Britany …	220	70000.00

DEPARTMENTS:

DEPT_ID_DEP	DEP_NAME	MANAGER_ID	LOC_ID
5	Software Group	30002	L0002
7	Design Team	30003	L0003

If we want to retrieve only the employee records that correspond to departments in the DEPARTMENTS table, we can use:

```
select * from employees

  where DEP_ID IN
```

```
  ( select DEPT_ID_DEP from departments );
```

Let's say we want to retrieve only the list of employees from a specific location. We do not have any location information in the EMPLOYEES table but the DEPARTMENTS table has a column called LOC_ID. So we can use a sub-query from the DEPARTMENTS table as input to the EMPLOYEE table query:

```
select * from employees

  where DEP_ID IN

   ( select DEPT_ID_DEP from departments

     where LOC_ID = 'L0002' );
```

Now let's retrieve the department ID and name for Empolyees who earn more than 70000:

```
select DEPT_ID_DEP, DEP_NAME from departments

 where DEPT_ID_DEP IN

  ( select DEP_ID from employees

    where SALARY > 70000 ) ;
```

2. Implicit join

Here we specify two tables in the query (but note we are not explicitly using the JOIN operator). The result is a full join (or cartesian join), because every row in the first table is joined with every row in the second table. If you examine the result set you will see more rows than in both tables individually.

```
select * from employees, departments;
```

We can use additional operands to limit the result set. In the following example we limit the result set to only rows with matching department IDs:

```
select * from employees, departments

  where employees.DEP_ID = departments.DEPT_ID_DEP;
```

Notice that in the where clause we pre-fixed the name of the column with the name of the table to fully qualify the column name since its possible that the different tables could have the some column names that are exactly the same. Since the table names can be sometimes long, we can use shorther aliases for table names as follows:

```
select * from employees E, departments D
```

```
  where E.DEP_ID = D.DEPT_ID_DEP;
```

Similarly, the column names in the select clause can be pre-fixed by aliases:

```
select E.EMP_ID, D.DEPT_ID_DEP

  from employees E, departments D

  where E.DEP_ID = D.DEPT_ID_DEP;
```

Let's say we want to see the department name for each employee:

```
select E.EMP_ID, D.DEP_NAME from

  employees E, departments D

  where E.DEP_ID = D.DEPT_ID_DEP
```

3. JOIN OPERATOR

We can explicitly use the JOIN operator to JOIN multiple tables in a single query. These include INNER JOIN and OUTER JOIN and are discussed in detail in additional lessons.