

Task 1: Çalışanları employees tablosundan seçin ve işe alınma tarihlerine (hire_date) göre azalan sırada sıralayın. Çalışan numarası (employee_no), ad (first_name), soyad (last_name) ve işe alınma tarihini (hire_date) seçin.

**SQL
sorgusu**
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT employee_no, first_name, last_name, hire_date FROM employees ORDER BY hire_date DESC;
```

Screenshot
(sağdaki
beyaz kutuya
yapıştırın)

The screenshot shows a SQL IDE interface. The top panel displays a script with the following SQL query:

```
--yorum satırıdır.  
--Çalışanları employees tablosundan işe alım tarihine göre listeleme  
  
SELECT employee_no, first_name, last_name, hire_date  
FROM employees  
ORDER BY hire_date DESC;
```

The bottom panel shows the results of the query in a table named 'employees 1'. The table has 9 rows and 5 columns: employee_no, first_name, last_name, and hire_date. The data is sorted by hire_date in descending order.

Grid	employee_no	first_name	last_name	hire_date
1	10,019	Lillian	Haddadi	1999-04-30
2	10,024	Suzette	Petty	1997-05-19
3	10,084	Tuval	Kalloufi	1995-12-15
4	10,022	Shahaf	Famili	1995-08-22
5	10,026	Yongqiao	Bertziss	1995-03-20
6	10,054	Mayumi	Schueller	1995-03-13
7	10,016	Kazuhito	Cappelletti	1995-01-27
8	10,008	Saniya	Kalloufi	1994-09-15
9	10,044	Mingsen	Casley	1994-05-21

The bottom status bar shows the table is in 'TRT | tr_TR | Writable' mode, with 'Smart Insert' enabled and a total of 6:25:186 records.

Task 2: 1960 yılı ve sonrasında doğmuş olan çalışanları employees tablosundan seçin. Ad (first_name), soyad (last_name) ve doğum tarihini (birth_date) seçin.

**SQL
sorgusu**
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT first_name, last_name, birth_date FROM employees WHERE birth_date >= '1960-01-01';
```

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot shows a SQL IDE interface. The top pane displays a script with two SQL queries. The first query is a comment: `--yorum satırıdır.`. The second query is: `--Çalışanları employees tablosundan işe alım tarihine göre listeleme`. The third query is: `SELECT employee_no, first_name, last_name, hire_date FROM employees ORDER BY hire_date DESC;`. The fourth query is: `--1960 yılı (dahil) ve sonrasında doğmuş olan çalışanları employees tablosundan listele`. The fifth query is: `SELECT first_name, last_name, birth_date FROM employees WHERE birth_date >= '1960-01-01';`. The bottom pane shows the execution results of the fifth query. It displays a table with 9 rows and 3 columns: first_name, last_name, and birth_date. The first row is: Bezalel, Simmel, 1964-06-02. The second row is: Duangkaew, Piveteau, 1963-06-01. The third row is: Patricio, Bridgland, 1960-10-04. The fourth row is: Eberhardt, Terkki, 1963-06-07. The fifth row is: Kazuhito, Cappelletti, 1961-05-02. The sixth row is: Ramzi, Erde, 1960-02-20. The seventh row is: Divier, Reistad, 1962-07-10. The eighth row is: Domenick, Tempesti, 1963-11-26. The ninth row is: Jeong, Reistad, 1960-08-09. The bottom status bar shows: TRT | tr_TR | Writable | Smart Insert | 8:90:368.

Grd	first_name	last_name	birth_date
1	Bezalel	Simmel	1964-06-02
2	Duangkaew	Piveteau	1963-06-01
3	Patricio	Bridgland	1960-10-04
4	Eberhardt	Terkki	1963-06-07
5	Kazuhito	Cappelletti	1961-05-02
6	Ramzi	Erde	1960-02-20
7	Divier	Reistad	1962-07-10
8	Domenick	Tempesti	1963-11-26
9	Jeong	Reistad	1960-08-09

Task 3: Adı "Ma" ile başlayan çalışanları employees tablosundan seçin. Ad (first_name), soyad (last_name) ve doğum tarihini (birth_date) seçin.

**SQL
sorgusu**
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';
```

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot displays a SQL IDE interface. The top pane, titled 'Script-1', contains the following SQL code:

```
--yorum satırıdır.  
--Çalışanları employees tablosundan işe alım tarihine göre listeleme  
  
SELECT employee_no, first_name, last_name, hire_date FROM employees ORDER BY hire_date DESC;  
  
--1960 yılı (dahil) ve sonrasında doğmuş olan çalışanları employees tablosundan listele  
  
SELECT first_name, last_name, birth_date FROM employees WHERE birth_date >= '1960-01-01';  
  
--Adı "Ma" ile başlayan çalışanları employees tablosundan listele.  
  
SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';
```

The bottom pane, titled 'employees 1', shows the execution results in a table grid. The table has columns: first_name, last_name, birth_date. The results are as follows:

	first_name	last_name	birth_date
1	Mary	Sluis	1953-11-07
2	Mayuko	Warwick	1952-12-24
3	Magy	Stamatiou	1956-02-26
4	Mayumi	Schuessler	1957-04-04
5	Margareta	Bierman	1960-09-06

The status bar at the bottom indicates '5 row(s) fetched - 0.001s, on 2025-05-27 at 18:52:37'.

Task 4: Kyiv, Lviv ve Dnipro'da yaşayan çalışanların ID'lerini employeeTerritories tablosundan seçin. Çalışan numarası (employee_no) ve şehir (city) seçin.

SQL sorgusu
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT employee_no, city FROM employeeTerritories WHERE city IN ('Kyiv', 'Lviv', 'Dnipro');
```

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot shows a SQL IDE interface. The top panel is a script editor with the following SQL code:

```
--yorum satırıdır.  
--Çalışanları employees tablosundan işe alım tarihine göre listeleme  
  
SELECT employee_no, first_name, last_name, hire_date FROM employees ORDER BY hire_date DESC;  
  
--1960 yılı (dahil) ve sonrasında doğmuş olan çalışanları employees tablosundan listele  
  
SELECT first_name, last_name, birth_date FROM employees WHERE birth_date >= '1960-01-01';  
  
--Adı "Ma" ile başlayan çalışanları employees tablosundan listele.  
  
SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';  
  
--Şehir olarak listele.  
  
SELECT employee_no, city FROM employeeTerritories WHERE city IN ('Kyiv', 'Lviv', 'Dnipro');
```

The bottom panel shows the results of the query in a grid format. The grid has two columns: employee_no and city. The data is as follows:

employee_no	city
10,002	Dnipro
10,004	Dnipro
10,005	Kyiv
10,006	Lviv
10,007	Kyiv
10,009	Dnipro
10,014	Dnipro
10,016	Kyiv

The bottom right panel shows a dictionary view for the employees table, with columns Value and Description. The data is as follows:

Value	Description
10002	Bezalel
10003	Parto
10004	Chirion

The bottom status bar shows: 34 row(s) fetched - 0.002s (0.001s fetch), on 2025-05-27 at 18:58:35.

Task 5: titles tablosundan "Engineer" pozisyonunu (title) işgal eden çalışanların sayısını hesaplayın. Sonuç sütununa "total engineers" adını verin.

**SQL
sorgusu**
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT COUNT(*) AS "total engineers" FROM titles WHERE title = 'Engineer';
```

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot shows a SQL IDE interface. The top panel, titled 'Script-1', contains a SQL query: `SELECT employee_no, first_name, last_name, hire_date FROM employees ORDER BY hire_date DESC;` followed by several comments in Turkish and more queries: `SELECT first_name, last_name, birth_date FROM employees WHERE birth_date >= '1960-01-01';`, `SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';`, `SELECT employee_no, city FROM employeeTerritories WHERE city IN ('Kyiv', 'Lviv', 'Dnipro');`, and finally the query for the task: `SELECT COUNT(*) AS "total engineers" FROM titles WHERE title = 'Engineer';`. The bottom panel, titled 'Results 1', shows the execution results in a table with one row: 'total engineers' with a value of 161. A 'Value' window on the right also displays the value 161. The status bar at the bottom indicates '1 row(s) fetched - 0.001s, on 2025-05-27 at 19:01:25'.

```
*<.sqlite> Script-1 x
SELECT employee_no, first_name, last_name, hire_date FROM employees ORDER BY hire_date DESC;

--1960 yılı (dahil) ve sonrasında doğmuş olan çalışanları employees tablosundan listele
SELECT first_name, last_name, birth_date FROM employees WHERE birth_date >= '1960-01-01';

--Adı "Ma" ile başlayan çalışanları employees tablosundan listele.
SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';

--Şehir olarak listele.
SELECT employee_no, city FROM employeeTerritories WHERE city IN ('Kyiv', 'Lviv', 'Dnipro');

--"Engineer" pozisyonunda çalışan sayısı
SELECT COUNT(*) AS "total engineers" FROM titles WHERE title = 'Engineer';

Results 1 x
Enter a SQL expression to filter results (use Ctrl+Space)
total engineers
1 161
Value x
161
Refresh Save Cancel 200 1 1 row(s) fetched - 0.001s, on 2025-05-27 at 19:01:25
```

Task 6: employeeTerritories tablosundan her şehir için çalışan sayısını görüntüleyin. Sadece 10'dan fazla çalışanı olan şehirleri dahil edin. Çalışan sayısını ve şehir adını görüntüleyin.

SQL sorgusu
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT city, COUNT(employee_no) AS employee_count FROM employeeTerritories GROUP BY city  
HAVING COUNT(employee_no) > 10;
```

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot displays a SQL IDE interface. The top pane shows a script with several SQL queries. The bottom pane shows the execution results of the query: `SELECT city, COUNT(employee_no) AS employee_count FROM employeeTerritories GROUP BY city HAVING COUNT(employee_no) > 10;`. The results are displayed in a table with two columns: `city` and `employee_count`. The table contains three rows: Dnipro (18), Kharkiv (17), and Odessa (11). The status bar at the bottom indicates that 3 rows were fetched in 0.002s.

```
--Adı "Ma" ile başlayan çalışanları employees tablosundan listele.  
SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';  
  
--Şehir olarak listele.  
SELECT employee_no, city FROM employeeTerritories WHERE city IN ('Kyiv', 'Lviv', 'Dnipro');  
  
--"Engineer" pozisyonunda çalışan sayısı  
SELECT COUNT(*) AS "total engineers" FROM titles WHERE title = 'Engineer';  
  
--employeeTerritories tablosundan her şehir için çalışan sayısını listele.  
SELECT city, COUNT(employee_no) AS employee_count FROM employeeTerritories GROUP BY city HAVING COUNT(employee_no) > 10;
```

city	employee_count
Dnipro	18
Kharkiv	17
Odessa	11

3 row(s) fetched - 0.002s (0.001s fetch) on 2025-05-27 at 19:04:50

Task 7: Çalışanların adını ve soyadını, şirketteki pozisyonları ile birlikte seçin. employees ve titles tablolarını kullanarak ad (first_name), soyad (last_name) ve pozisyonu (title) görüntüleyin.

**SQL
sorgusu**
(kodu
sağdaki
beyaz kutuya
yapıştırın)

SELECT e.first_name, e.last_name, t.title FROM employees e JOIN titles t ON e.employee_no = t.employee_no;

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot shows a SQLite IDE interface. The top pane displays a script with several SQL queries. The bottom pane shows the results of the last query, which is a JOIN of the employees and titles tables. The results are displayed in a table with columns: first_name, last_name, and title. The table contains 8 rows of data.

```
<sqlite> Script-1 x
SELECT first_name, last_name, birth_date FROM employees WHERE first_name LIKE 'Ma%';
--Şehir olarak listele.
SELECT employee_no, city FROM employeeTerritories WHERE city IN ('Kyiv', 'Lviv', 'Dnipro');
--"Engineer" pozisyonunda çalışan sayısı
SELECT COUNT(*) AS "total engineers" FROM titles WHERE title = 'Engineer';
--employeeTerritories tablosundan her şehir için çalışan sayısını listele.
SELECT city, COUNT(employee_no) AS employee_count FROM employeeTerritories GROUP BY city HAVING COUNT(employee_no) > 1;
--Çalışanların adını ve soyadını, şirketteki pozisyonları ile birlikte listele.
SELECT e.first_name, e.last_name, t.title FROM employees e JOIN titles t ON e.employee_no = t.employee_no;
```

	first_name	last_name	title
1	Georgi	Facello	Senior Engineer
2	Bezalel	Simmel	Staff
3	Parto	Bamford	Senior Engineer
4	Chirstian	Koblick	Engineer
5	Chirstian	Koblick	Senior Engineer
6	Kyoichi	Maliniak	Senior Staff
7	Kyoichi	Maliniak	Staff
8	Anneke	Preusig	Senior Engineer

Refresh Save Cancel Export data 200 134

134 row(s) fetched - 0.004s (0.004s fetch), on 2025-05-27 at 19:07:40

Task 8: Maaşı 50000 ile 60000 arasında olan çalışanların adını ve soyadını seçin. employees ve salaries tablolarını kullanarak ad (first_name), soyad (last_name) ve maaşı (salary) seçin.

SQL sorgusu
(kodu
sağdaki
beyaz kutuya
yapıştırın)

```
SELECT e.first_name, e.last_name, s.salary FROM employees e JOIN salaries s ON e.employee_no = s.employee_no  
WHERE s.salary BETWEEN 50000 AND 60000;
```

Screenshot
(kodu
sağdaki
beyaz kutuya
yapıştırın)

The screenshot shows a SQL IDE interface. The top pane displays a script with several SQL queries. The bottom pane shows the results of the last query, which is a table with 3 columns: first_name, last_name, and salary. The results are filtered to show only employees with a salary between 50,000 and 60,000. The table has 8 rows of data.

Grid	first_name	last_name	salary
1	Chirstian	Koblick	50,594
2	Chirstian	Koblick	52,119
3	Chirstian	Koblick	54,693
4	Chirstian	Koblick	58,326
5	Anneke	Preusig	52,255
6	Anneke	Preusig	53,747
7	Anneke	Preusig	56,032
8	Anneke	Preusig	58,299

Refresh Save Cancel Enter a SQL expression to filter results (use Ctrl+Space) 200 53

53 row(s) fetched - 0.000s, on 2025-05-27 at 19:10:17

TRT | tr TR | Writable | Smart Insert | 32 : ... 1359 :