Subquery as a single value

To find films whose `rental_rate` is at most 1.0 dollar from the maximum rate. We can do so in two steps:

- Compute the average rental rate using `AVG`.

  ```sql
  SELECT MAX(rental_rate) FROM film; // → 4.99
  ```

- Use this value in a query with a `WHERE` condition.

  ```sql
  SELECT title FROM film WHERE rental_rate ≥ 3.99;
  ```

We can do all of that in one query:

```sql
SELECT title, rental_rate FROM film
  WHERE rental_rate > (
    SELECT MAX(rental_rate) - 1.0 FROM film
  );
```

Copying tuples

INSERT allows you to specify tuples as a SELECT statement

```
INSERT INTO new_users (
  SELECT * FROM users WHERE active = true
);
```

```
INSERT INTO new_users (name) (
  SELECT name FROM users
);
```

Set Membership

In the clause `WHERE ... IN`, we can plug another `SELECT` query.

To list the customers that had a rental with return date 27/05/2005:

```
SELECT first_name, last_name
  FROM customer
  WHERE customer_id IN (
    SELECT customer_id
      FROM rental
      WHERE CAST(return_date AS DATE) = '2005-05-27'
  );
```

Subqueries in the FROM clause (1)

If we have the number of times every movie was rented:

```
SELECT title, count(rental_id) as times
  FROM film
  JOIN inventory using(film_id)
  JOIN rental using(inventory_id)
  GROUP BY film_id;
```

Subqueries in the **FROM** clause (2)

We can compute the average number of rentals per movie:

```sql
SELECT AVG(times) FROM (
  SELECT count(rental_id) as times
    FROM film
    JOIN inventory using(film_id)
    JOIN rental using(inventory_id)
    GROUP BY film_id
) AS rental_times;
```

Subqueries in the ***FROM*** *clause require using the* ***AS*** *keyword to name the resulting table.*

Common Table Expressions (CTEs)

A general way to partition a query into multiple steps is to use `WITH`, which makes a query be the input to another query:

```
WITH tmptable AS (
  SELECT ... FROM ...
    WHERE ...
)
SELECT * FROM tmptable, ...
  WHERE ...;
```

Common Table Expressions (CTEs): Example

Let’s rewrite the ”average number of times rented” query with CTEs:

```sql
WITH rental_times AS (
  SELECT count(rental_id) as times
    FROM film
    JOIN inventory using(film_id)
    JOIN rental using(inventory_id)
    GROUP BY film_id
)
SELECT AVG(times) FROM rental_times;
```